





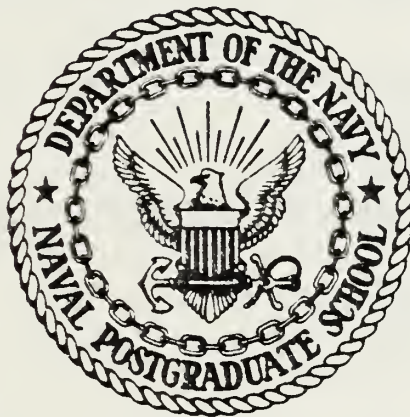






# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



INVESTIGATION OF NON-LINEAR ESTIMATION  
OF NATURAL RESONANCES  
IN TARGET IDENTIFICATION

by

Choong Y. Chong

December 1983

Thesis Advisor:

M. A. Morgan

Approved for public release; distribution unlimited.

T215125



## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS  
BEFORE COMPLETING FORM

1. REPORT NUMBER		2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Investigation of Non-linear Estimation of Natural Resonances in Target Identification		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1983	
7. AUTHOR(s)  Choong Y. Chong		6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		8. CONTRACT OR GRANT NUMBER(s)	
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1983	
		13. NUMBER OF PAGES 109	
		15. SECURITY CLASS. (of this report)  Unclassified	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Scattering Transient Response      Lease-Squares Method The New Signal Model                  Non-linear Optimization Entire Function Early Time Signal Prony's Algorithm			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  This investigation considers a non-linear technique for extracting natural resonances from transient electromagnetic scattering responses of radar targets. These natural resonances represent the complex poles of the target's transfer function in the Laplace transform s-plane. The advantage of their use in identification is their dependence only upon the			





Block 20 Contd.

geometry and composition of the target and not upon the aspect and polarization of the incident signal.

Based on recent theoretical research, a new signal model has been developed for describing the form of the transient scattering response. This form precludes the optimal use of previous methods for natural resonance extraction based on Prony's algorithm. In this effort, a modified least-squares approach is taken which can accommodate the actual form of the transient signal. The algorithm is tested using simulated signals with various noise levels and conclusions are drawn concerning the viability of the method.



Approved for public release; distribution unlimited.

INVESTIGATION OF NON-LINEAR ESTIMATION  
OF NATURAL RESONANCES  
IN TARGET IDENTIFICATION

by

Choong Y. Chong  
Lieutenant, Korean Navy  
B.S., Seoul National University, 1980

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

December 1983

1000000  
1000000  
1000000

## ABSTRACT

This investigation considers a non-linear technique for extracting natural resonances from transient electromagnetic scattering responses of radar targets. These natural resonances represent the complex poles of the target's transfer function in the Laplace transform  $s$ -plane. The advantage of their use in identification is their dependence only upon the geometry and composition of the target and not upon the aspect and polarization of the incident signal.

Based on recent theoretical research, a new signal model has been developed for describing the form of the transient scattering response. This form precludes the optimal use of previous methods for natural resonance extraction based on Prony's algorithm. In this effort, a modified least-squares approach is taken which can accommodate the actual form of the transient signal. The algorithm is tested using simulated signals with various noise levels and conclusions are drawn concerning the viability of the method.





## TABLE OF CONTENTS

I.	INTRODUCTION -----	8
II.	SCATTERING TRANSIENT RESPONSE SYSTEM -----	12
	A. TRANSIENT RESPONSE MECHANISM -----	12
	B. IMPULSE RESPONSE SIGNAL MODEL -----	14
	C. NOISE -----	17
III.	PROBLEMS WITH THE TRADITIONAL METHODS -----	18
	A. PRONY'S ALGORITHM -----	18
	B. HOUSEHOLDER ORTHOGONALIZATION METHOD -----	20
	C. EIGENVALUE METHOD -----	21
IV.	NON-LINEAR PARAMETER OPTIMIZATION APPROACH -----	23
	A. INTRODUCTION -----	23
	B. THE MODIFIED LEAST-SQUARES METHOD-----	25
V.	TEST RESULTS AND PERFORMANCE EVALUATION -----	27
	A. INTRODUCTION -----	27
	B. RESULTS -----	30
	C. PERFORMANCE EVALUATION -----	53
VI.	SUMMARY -----	56
	APPENDIX A. SYNTHETIC DATA GENERATION PROGRAM -----	58
	APPENDIX B. NON-LINEAR PARAMETER OPTIMIZATION PROGRAM---	76
	APPENDIX C. DATA SIGNAL PLOTS-----	101
	LIST OF REFERENCES -----	107
	INITIAL DISTRIBUTION LIST -----	109



# LIST OF TABLES

I.	SIMULATED SIGNAL 1 -----	27
II.	SIMULATED SIGNAL 2 -----	28
III.	SIMULATED SIGNAL 3 -----	28
IV.	PARAMETER OPTIMIZATION FOR SIGNAL 1 (NOISE FREE) -----	30
V.	PARAMETER OPTIMIZATION FOR SIGNAL 1 (SNR=30 dB) -----	31
VI.	PARAMETER OPTIMIZATION FOR SIGNAL 1 (SNR=15 dB) -----	32
VII.	PARAMETER OPTIMIZATION FOR SIGNAL 2 (NOISE FREE) -----	33
VIII.	PARAMETER OPTIMIZATION FOR SIGNAL 2 (SNR=30 dB) -----	34
IX.	PARAMETER OPTIMIZATION FOR SIGNAL 2 (SNR=15 dB) -----	35
X.	PARAMETER OPTIMIZATION FOR SIGNAL 3 (NOISE FREE) -----	36
XI.	PARAMETER OPTIMIZATION FOR SIGNAL 3 (SNR=30 dB) -----	38
XII.	PARAMETER OPTIMIZATION FOR SIGNAL 3 (SNR=15 dB) -----	40





## LIST OF FIGURES

5.1	Estimated $E(k)$ for Signal 1 of 15 dB SNR (32 data points from late time region) -----	42
5.2	Estimated $E(k)$ for Signal 1 of 15 dB SNR (48 data points from late time region) -----	43
5.3	Pole and Residue Plot for Signal 1 of Noise Free -----	44
5.4	Pole and Residue Plot for Signal 1 of 30 dB SNR -----	45
5.5	Pole and Residue Plot for Signal 1 of 15 dB SNR -----	46
5.6	Pole and Residue Plot for Signal 2 of Noise Free -----	47
5.7	Pole and Residue Plot for Signal 2 of 30 dB SNR -----	48
5.8	Pole and Residue Plot for Signal 2 of 15 dB SNR -----	49
5.9	Pole and Residue Plot for Signal 3 of Noise Free -----	50
5.10	Pole and Residue Plot for Signal 3 of 30 dB SNR -----	51
5.11	Pole and Residue Plot for Signal 3 of 15 dB SNR -----	52



## I. INTRODUCTION

This investigation considers a non-linear technique for extracting natural resonances from transient electromagnetic scattering responses of radar targets. These natural resonances represent the complex poles of the target's transfer function in the Laplace transform  $s$ -plane. The advantage of their use in identification is their dependence only upon the geometry and composition of the target and not upon the aspect and polarization of the incident signal.

Concepts and methodologies evolving from this idea have been developed based on the Singularity Expansion Method (SEM) by Baum [Ref. 1] in which the location of  $s$ -plane singularities characterizes a target. The direct extraction of poles (and residues) from the complex impulse response was initially attempted by Main and Moffat [Ref. 2]. These two early references spawned the idea of using natural resonances as the basis of non-cooperative target recognition (NCTR).

The application of the Householder orthogonalization method using a successive orthogonalization process and the eigenvalue method, that finds eigenvectors and eigenvalues, can determine the number and value of poles from the traditional transient signal model of a scatterer if there is a sufficiently low level of noise in the data [Ref. 3]. But both methods show poor quality of performance if we apply



them to the new signal model, even were there is no noise assumed in the signal data. This new signal model, which represents the electromagnetic scattering transient response of actual targets, was recently derived by Morgan [Ref. 4].

As considered in [Ref. 2], the relationship between a scattering target and the singularities being extracted from the natural complex resonance signal has been emphasized in the context of the number of poles and the accuracy of poles to ensure the unique relationship between an object and its image. Namely, the accuracy of pole extraction plays an important part in setting baselines to check consistency between a received signal and a set of singularities. Also, [Ref. 2] describes the modeling problem of a signal such that we express a transient response signal in terms of complex exponentials and the way it can be manipulated by Prony's algorithm. This signal model (termed here as the traditional model), was found not to represent the actual transient signal in the recent work by Davenport [Ref. 5]. In 1982, Manilha, in his work, commented that the new signal model might be partitioned into "early time signal" and "late time signal", each of which characterize the signal by the time instant at which the traditional model can be acceptable or not [Ref. 3]. This effort considers the development of a method that is capable of handling the "early time" whose time varying residues of exponentials do not become constant until the excitation of the incident field disappears. Such a signal





can not be adequately or accurately modeled by the traditional model.

It is the intention of this work to investigate a method that can handle the "early time signal" and show "Robustness" under relatively heavy noise pollution as well as improving the accuracy of the poles we are interested in evaluating. A non-linear parameter optimization approach, through the modified least-squares minimization method, has been tried. In particular, we have been interested in evaluating its performance in improving the accuracy of the poles extracted. The non-linear parameter optimization approach using iterations has contributed to improvement of accuracy and showed the possibility of application to the "early time signal" as was the final goal. We define the "new signal model" as a causal connection of the "early time signal" and "late time signal" throughout this paper. The new signal model will be presented in Chapter II, with both the description of the transient response mechanism and the way in which the signal model is constructed through the transient response mechanism. Chapter III briefly describes the problems associated with the traditional model, and with many of the current extraction methods in existence. Chapter IV is devoted to the description of the non-linear parameter optimization through a modified least-squares method and the way in which the algorithm is implemented. Chapter V presents test results of an attempt to use this approach for pole extraction and its performance characteristics. Chapter VI contains a summary



of the simulation results, the problems encountered, the potential problems expected, and the possibility of application of this method.

Computations were performed using the IBM 3033 system at the Naval Postgraduate School. The testing of concept feasibility was the major impetus behind this effort, with concerns for practical implementation in a "real-time" environment left for subsequent study. Considerations to the processing speed will not be addressed in this thesis. All the calculations were done in a double precision environment.





## II. SCATTERING TRANSIENT RESPONSE SYSTEM

### A. TRANSIENT RESPONSE MECHANISM

An electromagnetic wave incident upon a scattering body forces currents to be distributed such that Maxwell's equations and the corresponding boundary conditions are satisfied. These induced currents produce a scattered field that can be defined at any coordinate in free space if the expressions for those currents are made correctly. Changes in the angle of incidence, the incident wave shape, the polarization mode, and the target geometry and composition affect the results of the evaluation of the scattered transient response.

Basically, the singularities uniquely characterize a scatterer by their number and locations. The corresponding residues, which are interpreted as the weights of these poles, either can be viewed in the frequency domain or in the time domain.

As shown in [Ref. 3], equation (2.1) represents the form of the scattered signal impulse response of the target, in digital form

$$H(k) = H_0(kT) + \sum_{m=1}^{\infty} H_k \exp(kS_m T) \quad (2.1)$$

where  $H_0(t) = 0$  for  $t \geq T_0$ ,  $T$  is the sampling interval and the infinite sum represents the natural resonance response with complex  $S_m = \sigma_m + j\omega_m$  in the left-half plane. The poles constitute the parameters for NCTR of an object. The process



of extracting those parameters from the impulse response first collects the data points within a finite time period (time window).

The data collected is represented using equation (2.1) and equation (2.3) which partitions the signal into "early time" and "late time" components.

$$X(k) = \sum_{k=0}^{\infty} H(k) * \sum_{k=0}^{M-1} I(k) \quad (2.2)$$

$$X(k) = \sum_{k=0}^{T\emptyset-1} H(k) * \sum_{k=0}^{M-1} I(k) + \sum_{k=T\emptyset}^{\infty} H(k) * \sum_{k=\emptyset}^{M-1} I(k) \quad (2.3)$$

early time data                      late time data

where  $I(k)$  is the incident wave at  $t = kT$   
 $T$  is sampling interval in time.

We can write the equation (2.3) in the form of equation (2.4) or (2.5). Both of these expressions display the essence of the new signal model which will be discussed in the next section.

$$X(k) = \sum_{n=1}^N A_n(k) \exp(S_n k) \quad (2.4)$$

where  $A_n(k) = A_n$  for  $k \geq T\emptyset$

$$X(k) = E(k) + \sum_{n=1}^N A_n \exp(S_n k) \quad (2.5)$$

where  $E(k) = \emptyset$  for  $k \geq T\emptyset$

and  $A_n$  is constant for all  $k$



$T_0$  is the point at which the late time signal starts, so we consider that a time window might contain the data points ranging from the one before  $T_0$  to that after  $T_0$ . If we perform processing on a data window containing the data points which were from that early time region, we may lose some of the poles of natural resonance what are likely to be seen in the late time region. The data points after  $K_0$  presumably has all the information regarding poles, even though there are some extra poles that may be discriminated from the natural poles. These extra poles are introduced by the result of noise with relatively low SNR. Also, there is another risk to lose the natural poles having very high damping coefficients. There is no effective method yet found to compromise on having the point in which we can see all the poles with less extra poles under the higher SNR condition.

Meanwhile, the discrimination of  $E(k)$  appears to us as the quantity to be handled to give improvement in accuracy of the natural poles. The next section describes the new signal model based on these aspects and the transient response mechanism.

## B. IMPULSE RESPONSE SIGNAL MODEL

The late time impulse response can be expressed as a summation of exponentially damped sinusoids as in equation (2.6).

$$x_a(k) = \sum_{n=1}^N A_n \exp(S_n k), \quad k=0, T, 2T, \dots, (M-1)T \quad (2.6)$$



where  $T = \Delta t$ , and  $M$  is the number of sampling points. The signal model expressed in equation (2.6) is valid after the incident wave has completely illuminated a target and no forced response remains.

As the mechanism of a transient response system is strictly governed by the physics and orientation of the target and the attributes of the incident wave, we can write the equation (2.7) as an implicit form of equation (2.2), assuming we strike the target with a plane wave impulse.

$$X_b(k) = \sum_{k=\emptyset}^{\infty} H(k) \cdot \sum_{k=\emptyset}^{T\emptyset-1} [u(k) - u(k-T\emptyset)] \quad (2.7)$$

Combining both equations (2.6) and (2.7), equation (2.9) would be sufficient if we assume  $N$  poles are present.

$$\begin{aligned} X(k) &= X_a(k) + X_b(k) \\ X(k) &= \sum_{k=\emptyset}^{\infty} H(k) \cdot \sum_{k=\emptyset}^{T\emptyset-1} [u(k) - u(k-T\emptyset)] \\ &+ A_n \exp(S_n k), \quad k=\emptyset, T, 2T, \dots, (M-1)T \end{aligned} \quad (2.9)$$

We call the impulse response signal model, in the form of equation (2.9), an implicit form of the new signal model. A simpler form of equation (2.9) can be either of equations (2.10) or (2.11).

$$X(k) = E(k) + \sum_{n=1}^N A_n \exp(S_n k) \quad (2.10)$$

$$X(k) = \sum_{n=1}^N A_n(k) \exp(S_n k) \quad (2.11)$$





where  $A_n(k) = A_n$ , for  $k \geq T\emptyset$

In equation (2.1), the residues are time-varying until the time at which the augmented function  $E(k)$  in equation (2.10) becomes zero. Thereafter the  $A_n(k) = A_n$  (constant).

To begin with equation (2.10), it is necessary to define a new independent variable as a parameter whose time behavior is not predictable along the positive time axis. It is convenient to define this as a set of variables. By regarding them as independent, the equation (2.11) would be in the form of the equation (2.12) in the discrete digital data processing sense.

$$X(k) = E(k) + \sum_{n=1}^N A_n \exp(S_n k) \quad (2.12)$$

$$X(k) = \sum_{n=1}^{T\emptyset-1} e_n(k) + \sum_{n=1}^N A_n \exp(S_n k) \quad (2.13)$$

assuming  $e_n(k) = E_n \delta(k-n)$

The equation (2.14) is expressed in terms of a positive counting sequence in  $k$ . As in the equation, we might evaluate the equation (2.13) if we know the exact number of poles. But we know that it is impossible because we are in the region of the early time. So we write the equation (2.14) as a time shifted version such as the equation that has meaning from the observability viewpoint.

$$X(k) = \sum_{n=1}^{T\emptyset-1} e_n(k) + \sum_{n=1}^N A_n \exp(S_n k) \quad (2.14)$$



where  $k=T\emptyset+m, T\emptyset+m-1, \dots, T\emptyset+1, T\emptyset, T\emptyset-1, \dots, T\emptyset, \emptyset$   
for  $m \geq 0$

Here, the examples of the new signal model are presented in Appendix C, in the form of a decomposed signal. This synthetically generated signal is written in terms of  $e_n(k)$ 's and the sum of exponentials. In Chapter V, we present the decomposed signal from the synthetically generated data signal and that obtained by constructing the  $E(k)$  as the results of the non-linear parameter optimization processing.

### C. NOISE

White Gaussian noise is assumed throughout this work. By introducing the noise into the new impulse signal model, we call the following equation the modeling function of a transient response system and the complete and general form of the new signal model.

$$X(k) = N(k) + \sum_{n=1}^{\infty} e_n(k) + \sum_{n=1}^N A_n \exp(S_n k) \quad (2.15)$$



### III. PROBLEMS WITH THE TRADITIONAL METHODS

In this chapter, a brief description of problems with the classic Prony's method, the Householder orthogonalization technique and the Eigenvalue method are provided. Each of these methods computes the location of poles in the s-plane but only the last two methods can estimate the number of poles prior to the discrimination of the poles.

#### A. PRONY'S ALGORITHM

Equation (2.3) is interpreted as the following difference equation assigning the values of the real part of the signal at specified sampling points in its left side of the expression.

$$\begin{aligned} X(\emptyset) &= \sum_{n=1}^N A_n \\ X(k) &= \sum_{n=1}^N A_n z_n^k \\ &\vdots \end{aligned} \tag{3.1}$$

$$X((M-1)k) = \sum_{n=1}^N A_n z_n^{(M-1)k}$$

And a polynomial equation (3.2) that has the same roots  $z_n$

$$\sum_{n=1}^N a_n z^n = \emptyset \tag{3.2}$$



can be combined with equation (3.1) to yield the Prony's difference equation in the form of equation (3.3)

$$\sum_{n=1}^N a_n X(n+k) = 0, \quad \text{for } k=0, T, 2T, \dots, (M-1) T \quad (3.3)$$

Then, we write equation (3.4) from equation (3.3) as

$$\sum_{n=1}^N a_n X(n+k) = \sum_{n=1}^{N-1} a_n X(n+k) + a_N x(n+k) = 0 \quad (3.4)$$

Using matrix notation, the equation (3.3) may be written as equation (3.5) or (3.6).

$$\sum_{n=1}^{N-1} a_n X(n+k) = -a_N X(n+k) \quad (3.5)$$

Let  $a_N = 1$ , then equation (3.5) becomes as

$$X_{N-1} A_{N-1} = x_N \quad (3.6)$$

where  $X_{N-1}$  is N by N circulant matrix of sampled data

$A_{N-1}$  is N by 1 Prony's coefficients matrix

$x_N$  is N by 1 row matrix of sampled data set

$$[x_N, x_{N+1}, \dots, x_{2N-1}]$$

The Prony's coefficient  $a_n$ 's are to be calculated by making use of the characteristics of the circulant matrix X.

$$A_{N-1} = (X_{N-1} X_{N-1})^{-1} X_{N-1} x_N \quad (3.7)$$

As described in the above, this method can be applied after the number of poles are known to us, then the value of the  $S_n$ 's are to be found from the equation (3.8).





$$S_n = \text{LN}(Z_n)/k, \text{ for } k \geq 2N \quad (3.8)$$

## B. HOUSEHOLDER ORTHOGONALIZATION METHOD

A general expression of equation (3.2) may be represented by equation (3.9) assuming the number of poles are not known.

$$a_0 + a_1 z^1 + \dots + a_{N'} z^{N'} = 0 \quad (3.9)$$

where  $N'$  is the unknown variable

Then, the equation (3.9) is to be satisfied under the condition that there are enough sampled data points  $m \geq 2N$ .

$$a_0 x_0 + a_1 x_1 + \dots + a_{N'} x_{N'} = 0 \quad (3.10)$$

where the  $N$  by  $1$  sampled data matrix is defined as

$$x_i = [X(i), X(i+2), \dots, X(M-N-i+1)]$$

Application of the successive orthogonalization through the Gram-Schmit process would produce the orthogonal vector set.

$$O = \{o_0, o_1, \dots, o_{N'}\}$$

$$\text{where } o_n = x_n - \sum_{i=0}^{n-1} \langle x_n, o_i \rangle o_i \quad (3.11)$$

If we set the  $o_0$  to be  $1$ , then the  $N$  by  $1$   $x_n$  matrix is going to be in the form of equation (3.11).

$$x_n = o_n + \sum_{i=0}^{n-1} \langle x_n, o_i \rangle o_i \quad (3.12)$$

So that the equation (3.12) holds as  $X=OM$ . In the above,  $M$  is the  $2$ -D matrix of multiplication factors whose diagonal



elements are all 1's. Here, an orthogonal vector  $o_1$  makes the corresponding data set  $x_i$  to be orthogonal to all the previous vectors  $x_0 - x_{i-1}$ . If we have had all the  $x$  vectors during  $i-1$  times of successive orthogonalization process such that there is a non-zero orthogonal vector. We have to receive continuously the next data set  $x_{i+1}$  and check whether the orthogonal vector in the next step is zero or under a given threshold. If it vanishes, we can say that we have  $(i-1)$  poles. Our test against noise polluted data signal using the Householder method showed the inability of handling the early time signal as well as the data being heavily polluted.

### C. EIGENVALUE METHOD

Again, from the equation (3.10), we can rewrite this equation as in the form of equation (3.14).

$$X_N Z_N = 0 \quad (3.14)$$

$$\text{where } X_N = [x_0 | x_1 | \dots | x_N]$$

The eigenvector can be derived from the equation (3.15) by doing some mathematical manipulations of equation (3.14).

$$X_N^T X_N A_N = X_N^T X_N E_N = 0$$

From this equation, we find the eigenvalues correspond to eigenvectors to see if there is any eigenvalue approaching zero. If there is one zero or under threshold, we also consider the number of poles as  $N'-1$ . Even both the



Householder orthogonalization method and the Eigenvalue method can provide the means of calculating the number of poles, their algorithm (tending to fit the data points to the traditional signal model) showed the lack of generality.



#### IV. NON-LINEAR PARAMETER OPTIMIZATION APPROACH

##### A. INTRODUCTION

In 1963, Marquardt suggested an algorithm for the least-squares estimation of non-linear parameters [Ref. 6] when highly accurate parameter values are required. As we can see in the signal model in equation (2.14), that new signal model has multiple parameters that are functions of time. Here, we rewrite that equation again in the form of equation (4.1).

$$X(k) = N(k) + \sum_{n=k}^{T\emptyset-1} e_n(k) + \sum_{n=1}^N A_n \exp(S_n k) \quad (4.1)$$

where  $e_n(k) = \emptyset$ , for  $k \geq T\emptyset$

We are not interested in the value of  $e_n(k)$  itself, but include their estimations in order to contribute to the accuracy of  $S_n$ 's and  $A_n$ 's.

In other words, the extrapolation of the sum of exponentials, which is assumed to have all the poles, to points in the early time region may provide more accurate poles and residues simultaneously.

The major advantage is the fact that we can make use of the high SNR data signal in the early time region with the basic information of poles that were derived in the late time region.





We will be processing on a data window whose vector length is to be increased one by one by moving the first element of that vector toward the time-origin point. The optimization process evaluates the normal equation to find the local optimized set and finds the global optimized set as its final goal.

Let us define an ERROR function as in equation (4.3)

$$\text{ERROR}\{A_n, S_n, E(k)\} = \sum_{n=1}^{N-1} [X'(k) - X(k)]^2 \quad (4.3)$$

where  $X'(k)$  is a measured data point

To have minimized the least-squares error at every instant of measured time, the ERROR function has to be minimized by obtaining the global set of parameters being optimized. We also can write equation (4.3) in a more concrete manner as in equation (4.4).

$$\text{ERROR}\{X\} = \sum_{k=\emptyset}^{M-1} [X'(k) - X(k)]^2 = \sum_{k=\emptyset}^{M-1} e(k)_n^2 \quad (4.4)$$

where  $X = [E(k), A_n, S_n]$

Non-linear parameters such as  $e_n(k)$  must be obtimized in a way such that the more accurate values of  $S_n$  have to be calculated as we increase the number of non-linear parameters  $e_n$ . Users may define the numbers of parameter  $e_n$ 's that are at least equal to or greater than the number of data points in the early time region. Now it is our task to find out the optimized value of  $A_n, S_n$ , and  $e_n$  through optimization processing, either in the global sense as it is represented in the equation (4.5) or using the normal equation (4.6).



$$\partial \text{ERROR}(X) / \partial X = 0 \quad (4.5)$$

$$\partial e_n(k) / \partial X(k) = 0 \quad (4.6)$$

where  $k=0, T, 2T \dots, (M-1)K$

$n=1, 2, \dots, N$

With using the normal equation, as it is shown in the (4.6), a global set of non-linear set of parameters can be obtained through an iterative evaluation and fitting process. In our simulation work, the modified least-squares method was the basis of the equation (4.6).

#### B. THE MODIFIED LEAST-SQUARES METHOD

The error function is rewritten as in the expression of the equation (4.7).

$$\text{ERROR}\{k, x\} = X'(k) - X(k, x_n^k) \quad (4.7)$$

where  $X'(k)$  is the  $k$ -th sampled data

$X(k, x)$  is the model function

$x_n^k$  is a vector containing the  $k$ -th sampled

Let us define  $x_n^0$  to be an initial estimated value of  $x$ , then a sequence of approximations to the optimized value is to be generated by the equation (4.8).

$$x^{m+1} = x^m - [a_m D_m + J_m^T J_m]^{-1} J_m^T \text{ERROR}(x^m) \quad (4.8)$$

where  $J_m$  is the numerical Jacobian matrix evaluated

$m$  is iteration number of successive optimization

$D_m$  is a diagonal matrix equal to the diagonal of



$$J_m^T J_m$$

$a_m$  is a Marquardt parameter

The number of total iterations can be controlled by the threshold which is defined as in the equation (4.9).

$$d = x^{m+1} - x^m \quad (4.9)$$



## V. TEST RESULTS AND PERFORMANCE EVALUATION

### A. INTRODUCTION

In order to establish the ability of the program listed in Appendix B to extract the poles and residues correctly, three different simulated signals, each of which is polluted with infinite, 30 dB and 15 dB SNR, were created by the synthetic signal data generation routine. These 3 sets of signal data were chosen to span a range of possible situations.

In the context of the transient signals and additive stationary Gaussian noise being used, the definition of SNR is in terms of a ratio of energy quantities intergrated over the entire 20 nsec time window.

TABLE I  
SIMULATED SIGNAL 1

<u>RESIDUES</u>	<u>POLES</u> (Nep. GHZ)
$1.0 + i1.0$	$-1.0 + i1.0$
$1.0 - i1.0$	$-1.0 - i1.0$

Simulated signal 2 consists of 2 sets of pairs of complex conjugate poles and residues, which is an extrapolation of the simulated signal 1, using its parameters. Simulated signal 3 has 3 sets of pole-residue pairs, extrapolating from signal 2.

There are 5 options in the signal generation program in choosing a particular function of  $E(k)$ . In this simulation work, a trapezoidal wave form of  $E(k)$  was used with  $T_0 = 0.42$  nsec. So that the unknown parameters that are to be augmented





at every processing step will be a maximum of 10 plus 4 times the number of poles at the final processing stage, when we have 512 data points within a 20 nsec time window.

TABLE II  
SIMULATED SIGNAL 2

<u>RESIDUES</u>	<u>POLES</u> (Nep. GHZ)
$1.0 + i1.0$	$-1.0 + i1.0$
$1.0 - i1.0$	$-1.0 - i1.0$
$0.5 + i0.5$	$-2.0 + i2.0$
$0.5 + i0.5$	$-2.0 - j2.0$

TABLE III  
SIMULATED SIGNAL 3

<u>RESIDUES</u>	<u>POLES</u> (Nep. GHZ)
$1.0 + i1.0$	$-1.0 + i1.0$
$1.0 - i1.0$	$-1.0 - i1.0$
$0.5 + i0.5$	$-2.0 + i2.0$
$0.5 - i0.5$	$-2.0 - i2.0$
$0.25 + i0.25$	$-3.0 + i3.0$
$0.25 - i0.25$	$-3.0 - i3.0$

The results of varying the number of additional data points from zero (all the data points are from late time region) to ten extra data points (10 extra points are from early time region are added to the points of late time region) are contained in Table IV through XII. These nine



cases correspond to the 3 synthetic signals, each having 3 additive noise levels. The 3 synthetic signals are plotted in Appendix C. In the next section, the accuracy of pole extractions, as indicated in the tables are displayed by way of graphical pole maps. In addition, reconstructed waveform obtained from the parameter extractions are compared graphically to the original waveforms.



TABLE IV

## PARAMETER OPTIMIZATION FOR SIGNAL 1 (NOISE FREE)

## B. RESULTS

TARGET TYPE:TGT-1  
 WAVEFORM TYPE:FCNFTR  
 CONTACT DATE:DEC 15  
 FILE NAME:FILE002  
 NUMB. OF POLE: 2

 TABLE CF RESIDUES AND POLES  
 =====

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000

## RESULTS CF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99996610	1.00001592	-0.99994705	0.99998971
2	0.99996563	-1.00001375	-0.99994777	-0.99998971

## RESULTS CF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000251	0.9999667	-1.0000025	1.00000085
2	1.00000251	-0.9999667	-1.00000026	-1.00000085

## RESULTS CF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00001308	1.00000329	-1.00001536	0.99999999
2	1.00001311	-1.00000330	-1.00001545	-1.00000002

## RESULTS CF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99995213	1.00003040	-0.99979425	0.99998208
2	0.99995210	-1.00003047	-0.99979441	-0.99998213

## RESULTS CF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000036	1.00001011	-1.000000472	0.99999818
2	1.00000026	-1.00001041	-1.000000540	-0.99999838

## RESULTS CF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000457	0.99999062	-0.99999674	1.00000165
2	1.00000453	-0.99999067	-0.99999735	-1.00000168



TABLE V  
PARAMETER OPTIMIZATION FOR SIGNAL 1  
(SNR=30 dB)

TARGET TYPE:TGT-1  
WAVEFORM TYPE:PCN30TR  
CONTACT DATE:CEC 15  
FILE NUME:FILE003  
NUMB. CF POLE: 2

TABLE CF RESIDUES AND POLES  
=====

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000

RESULTS CF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01000000	0.92346346	-0.77676493	1.00870076
2	1.03315188	-1.10738236	-1.30247623	0.99197436

RESULTS CF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01070336	1.00360284	-1.00951167	0.99979304
2	1.01070339	-1.00360275	-1.00951163	0.99979306

RESULTS CF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01066217	1.00367655	-1.00953849	0.99977751
2	1.01066632	-1.00367409	-1.00953782	-0.99977735

RESULTS CF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01059740	1.00357123	-1.00930702	0.99978580
2	1.01059679	-1.00357126	-1.00930751	0.99978574

RESULTS CF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01067940	1.00362571	-1.00950587	0.99978686
2	1.01068033	-1.00362636	-1.00950588	-0.99978699

RESULTS CF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01069621	1.00360379	-1.00950576	0.99979175
2	1.01069609	-1.00360399	-1.00950630	-0.99979177





TABLE VI

PARAMETER OPTIMIZATION FOR SIGNAL 1  
(SNR=15 dB)

TARGET TYPE:TGT-1  
WAVEFORM TYPE:HDN15TR  
CONTACT DATE:CEC 20  
FILE NUME:FILE004  
NUMB. OF POLE: 2

TABLE CF RESIDUES AND POLES  
=====

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000

RESULTS CF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.56047582	1.35923892	-0.57890084	0.99886304
2	1.59727710	-0.80336947	-1.57013325	-0.91419374

RESULTS CF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.06057059	1.02070489	-1.05306345	0.99869879
2	1.06057523	-1.02071795	-1.05306276	-0.99869878

RESULTS CF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.06053386	1.02072970	-1.05304404	0.99868990
2	1.06052818	-1.02073354	-1.05304091	-0.99869088

RESULTS CF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.06051459	1.02073547	-1.05295266	0.99873653
2	1.06054674	-1.02072794	-1.05313212	-0.99864419

RESULTS CF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.02449720	1.09889916	-0.66485231	1.01775339
2	1.07048258	-1.10030318	-1.63907420	-0.93004124

RESULTS CF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.03166928	1.11984550	-1.56675813	0.92761091
2	1.03351816	-1.07048377	-0.68289631	-1.02160947



TABLE VII

PARAMETER OPTIMIZATION FOR SIGNAL 2  
(NOISE FREE)

TARGET TYPE:TGT-2  
WAVEFORM TYPE:HCNFTF  
CONTACT DATE:DEC 15  
FILE NAME:FILE002  
NUMB. CF POLE: 4

## TABLE CF RESIDUES AND POLES

=====

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000

## RESULTS CF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000054	0.99998637	-1.00000054	1.00000162
2	1.00000019	-0.99998792	-0.99998223	-1.00000180
3	0.499993635	0.500005618	-2.000005518	1.999995365
4	0.499995131	-0.500004431	-2.00024952	-1.999994115

## RESULTS CF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99996513	0.99545512	-0.99792594	0.99998185
2	0.99996405	-0.99549467	-0.99792606	-0.99998794
3	0.47735342	0.48850313	-1.95118748	1.99896225
4	0.47735838	-0.48850442	-1.95306779	-1.99850918

## RESULTS CF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99996256	0.99545224	-0.99793418	0.99998541
2	0.99996321	-0.99545228	-0.99791438	-0.99998413
3	0.47737187	0.48848088	-1.95232122	1.99855642
4	0.47736975	-0.48847526	-1.95190026	-1.99893520

## RESULTS CF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99996743	0.99545467	-0.99793003	0.99998501
2	0.99996717	-0.99549457	-0.99792354	-0.99998468
3	0.47739612	0.48847662	-1.95179495	1.99874576
4	0.47739443	-0.48847686	-1.95248987	-1.99875751

## RESULTS CF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01514809	0.98882407	-1.12954971	1.01650809
2	1.00949031	-0.98875294	-0.88470066	-0.99067696
3	0.78175371	0.36507609	-2.03189450	2.04744671
4	0.14319333	-0.56814961	-1.95892411	-1.92582259

## RESULTS CF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.02706356	0.98982102	-1.16328021	1.01520699
2	1.02053154	-0.99088746	-0.86815870	-0.99077552
3	0.34656670	0.59967646	-1.95544269	1.95297875
4	0.61555028	-0.37246541	-2.00784742	-2.04312205



TABLE VIII

PARAMETER OPTIMIZATION FOR SIGNAL 2  
(SNR=30 dB)

TARGET TYPE: TGT-2  
WAVEFORM TYPE: FDN30TR  
CONTACT DATE: DEC 15  
FILE NUMB: FILE003  
NUMB. CF POLE: 4

TABLE CF RESIDUES AND POLES  
=====

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000

RESULTS CF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.999412945	0.99056805	-0.92918755	0.99302604
2	0.99948280	-0.99069937	-1.06492363	-1.00865606
3	0.00000010	0.56023544	-1.96891695	1.90034350
4	0.87406384	-0.32304349	-2.04566871	-2.05396332

RESULTS CF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99906513	0.99549512	-0.99792594	0.99998185
2	0.99906405	-0.99549567	-0.99792606	-0.99998794
3	0.47735342	0.48850313	-1.95118748	1.99896225
4	0.47735838	-0.48850442	-1.95306779	-1.99850918

RESULTS CF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99906256	0.99549224	-0.99793418	0.99998541
2	0.99906321	-0.99549228	-0.99791438	-0.99998413
3	0.47737187	0.48848088	-1.95232122	1.99855642
4	0.47736977	-0.48847526	-1.95190026	-1.99893520

RESULTS CF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99906743	0.99549467	-0.99793003	0.99998501
2	0.99906717	-0.99549457	-0.99792354	-0.99998468
3	0.47739612	0.48847662	-1.95179495	1.99874576
4	0.47739443	-0.48847686	-1.95248987	-1.99875751

RESULTS CF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01514809	0.98882407	-1.12954971	1.01650809
2	1.00949031	-0.98875294	-0.88470066	-0.99067696
3	0.78175371	0.36507609	-2.03189450	2.04744671
4	0.14319333	-0.56814961	-1.95892411	-1.92582259

RESULTS CF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.02706396	0.98982102	-1.16328021	1.01920699
2	1.02053154	-0.99088746	-0.86815870	-0.99077552
3	0.34656670	0.59967646	-1.95544269	1.95297875
4	0.61555028	-0.37246541	-2.00784742	-2.04312205





TABLE IX

PARAMETER OPTIMIZATION FOR SIGNAL 2  
(SNR=15 dB)

TARGET TYPE: TGT-2  
WAVEFORM TYPE: FDN15TR  
CONTACT DATE: DEC 15  
FILE NAME: FILE004  
NUMB. CF POLE: 4

TABLE CF RESIDUES AND POLES

=====

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000

RESULTS CF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96691205	0.96731225	-0.97658042	0.99822948
2	0.96692603	-0.96731915	-0.97544458	-0.99811586
3	0.52307777	0.22152280	-1.54313287	2.10614539
4	0.00000010	-0.42428522	-1.38794994	-1.83372419

RESULTS CF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96511311	0.96627888	-0.99390301	1.00027364
2	0.96526467	-0.96632064	-0.99566500	-0.99679353
3	0.00000010	0.49329107	-1.46905167	1.85021464
4	0.55450458	-0.13648901	-1.66572130	-2.12454838

RESULTS CF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96128516	0.96679315	-0.98935769	0.99965070
2	0.96144733	-0.96682559	-0.96226170	-0.99687448
3	0.00000010	0.54101355	-1.50069226	1.85232329
4	0.50251383	-0.09527370	-1.60980060	-2.13416731

RESULTS CF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.06294479	0.93763540	-1.25949424	1.04372728
2	1.01484084	-0.94095523	-0.77277970	-0.98546978
3	0.62674427	0.16116490	-1.75036751	2.11201876
4	0.00000010	-0.51970056	-1.59036824	-1.84519866

RESULTS CF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96304511	0.96782052	-0.98449408	0.99905219
2	0.96312358	-0.96782270	-0.96540893	-0.99752145
3	0.00000010	0.39719908	-1.36627993	1.83147922
4	0.54632361	-0.25136131	-1.56644065	-2.10011869

RESULTS CF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00071067	0.96985420	-1.20987398	1.02332284
2	1.00285344	-0.97024550	-0.81824047	-0.98822709
3	0.57607620	0.19510021	-1.66621998	2.10722391
4	0.00000010	-0.50028280	-1.53834141	-1.84201571





TABLE X

PARAMETER OPTIMIZATION FOR SIGNAL 3  
(NOISE FREE)

TARGET TYPE: TGT-3  
WAVEFORM TYPE: FCNFTF  
CONTACT DATE: DEC 15  
FILE NUMB: FILE004  
NUMB. CF POLE: 6

## TABLE CF RESIDUES AND POLES

PAIR #	RES.-REAL	RES.-IMAG	PCLE.-REAL	PCLE.-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000
5	0.25000000	0.25000000	-3.00000000	3.00000000
6	0.25000000	-0.25000000	-3.00000000	-3.00000000

## RESULTS CF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	PCLE.-REAL	PCLE.-IMAG
1	1.00000029	0.99999951	-1.00000013	1.00000000
2	0.99999965	-1.00000047	-0.99999935	-0.99999999
3	0.50000021	0.49999977	-1.99999933	1.99999965
4	0.49999957	-0.50000051	-2.00000029	-2.00000029
5	0.25000049	0.25000087	-2.99876241	2.99999506
6	0.24999768	-0.24999958	-3.00125324	-2.99999619

## RESULTS CF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	PCLE.-REAL	PCLE.-IMAG
1	1.00000033	1.00000006	-0.99999964	0.99999972
2	0.99999970	-0.99999990	-1.00000035	-1.00000028
3	0.49999957	0.49999976	-2.00000082	1.99999976
4	0.50000019	-0.49999970	-1.99999827	-2.00000017
5	0.24999312	0.25000063	-2.99929864	2.99995040
6	0.25000768	-0.24999482	-3.00067868	-3.00005033

## RESULTS CF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	PCLE.-REAL	PCLE.-IMAG
1	1.00000031	1.00000004	-0.99999929	0.99999999
2	1.00000008	-0.99999999	-1.00000008	-1.00000002
3	0.50000029	0.49999978	-2.00000158	2.00000057
4	0.50000073	-0.49999913	-1.99999746	-1.99999986
5	0.24999713	0.25000137	-3.00006822	3.00000834
6	0.25000306	-0.24999453	-2.99952441	-2.99999365



RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99999967	1.00000006	-0.99999978	0.99999979
2	0.99999949	-1.00000005	-1.00000003	-1.00000016
3	0.50000350	0.49999519	-2.00000411	-2.00000118
4	0.50000357	-0.49999515	-2.00000565	-2.00000116
5	0.25002351	0.25003725	-3.01254931	-2.99994026
6	0.249995017	-0.25002880	-2.98710687	-2.99987758

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	0.99999980	-0.99999966	1.00000026
2	1.00000001	-1.00000010	-1.00000031	-0.99999975
3	0.50000063	0.49995686	-2.00000881	-1.99969134
4	0.49999755	-0.50003918	-1.99998120	-2.00030891
5	0.250066787	0.25234001	-3.00656519	-3.00052848
6	0.24913768	-0.24765392	-2.99334543	-2.99547896

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000004	0.99999995	-1.00000147	1.00000006
2	0.99999994	-1.00000008	-0.99999854	-0.99999994
3	0.50000006	0.50000023	-2.00001924	-1.99999846
4	0.50000019	-0.50000011	-1.99998131	-2.00000154
5	0.25000013	0.25000022	-2.99992168	-2.99997583
6	0.25000024	-0.25000077	-3.00008161	-3.00002382



TABLE XI  
PARAMETER OPTIMIZATION FOR SIGNAL 3  
(SNR=30 dB)

TARGET TYPE:TGT-3  
WAVEFORM TYPE:HON30TR  
CONTACT DATE:DEC 15  
FILE NUMB:FILE004  
NUMB. OF POLE: 6

TABLE OF RESIDUES AND POLES  
=====

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.000000000	1.000000000	-1.000000000	1.000000000
2	1.000000000	-1.000000000	-1.000000000	-1.000000000
3	0.500000000	0.500000000	-2.000000000	2.000000000
4	0.500000000	-0.500000000	-2.000000000	-2.000000000
5	0.250000000	0.250000000	-3.000000000	3.000000000
6	0.250000000	-0.250000000	-3.000000000	-3.000000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99145389	0.98947089	-0.99318094	1.00001443
2	0.99145320	-0.98947086	-0.99310558	-1.00001237
3	0.42807658	0.52733218	-1.99204259	1.97341708
4	0.42846454	-0.52679082	-1.92072086	-1.99797545
5	0.11656813	0.23647251	-2.27124794	2.83611689
6	0.12939770	-0.21591933	-2.24119798	-3.06874116

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99147340	0.98938441	-0.99323692	1.00002118
2	0.99147541	-0.98938631	-0.99299518	-1.00001483
3	0.42927584	0.52783363	-1.96123606	1.86208680
4	0.42938721	-0.52783004	-1.96042724	-1.98830969
5	0.13113198	0.20433430	-2.22870752	3.07350223
6	0.10954559	-0.24676563	-2.28661642	-2.83356831

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99149793	0.98938042	-0.99300311	1.00001643
2	0.99149653	-0.98937734	-0.99323733	-1.00002228
3	0.42898254	0.52744729	-1.95708423	1.98692809
4	0.42897978	-0.52746013	-1.96340662	-1.98537735
5	0.13728153	0.20601283	-2.24720729	3.07453032
6	0.10518882	-0.24183713	-2.27543055	-2.83274078



# RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99181508	0.98944282	-0.99314260	1.00003060
2	0.99181556	-0.98944354	-0.99332473	-1.00003549
3	0.43071414	0.52480605	-1.97634267	1.98151201
4	0.43071211	-0.52479742	-1.93934530	-1.99173878
5	0.15055056	0.25270674	-2.28536097	2.85503619
6	0.10533241	-0.20048107	-2.19638655	-3.07720611

# RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.98171120	0.98981745	-0.99074872	0.99947232
2	0.98171187	-0.98981744	-0.99060276	-0.99949877
3	0.30805732	0.49840155	-1.88871352	1.87285997
4	0.31532825	-0.51226642	-1.56132982	-2.02127811
5	0.11572596	0.13443903	-1.99741621	3.12706630
6	0.07739011	-0.19217344	-1.89148981	-2.85558132

# RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99337417	0.99015792	-0.99418591	1.00008890
2	0.99337150	-0.99015803	-0.99369290	-1.00007652
3	0.44239542	0.52004030	-1.95770753	1.99113003
4	0.44239753	-0.52004768	-1.96547995	-1.98746271
5	0.15478007	0.23652063	-2.41828246	2.87099551
6	0.15377770	-0.23805063	-2.48497797	-3.06515871





TABLE XII  
PARAMETER OPTIMIZATION FOR SIGNAL 3  
(SNR=15 dB)

TARGET TYPE:TGT-3  
WAVEFORM TYPE:FCN15TR  
CONTACT DATE:DEC 15  
FILE NUME:FILE004  
NUMB. CF POLE: 6

TABLE CF RESIDUES AND POLES  
=====

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000
5	0.25000000	0.25000000	-3.00000000	3.00000000
6	0.25000000	-0.25000000	-3.00000000	-3.00000000

RESULTS CF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96825765	0.94693282	-0.96776591	1.00067712
2	0.96825659	0.94693466	-0.96776560	-1.00067721
3	0.25778800	0.55401616	-1.81344349	1.95442546
4	0.25778439	0.55401729	-1.81338128	-1.95442985
5	0.00000010	0.30602226	-2.40607412	2.86369297
6	0.00000010	-0.30602324	-2.40740413	-2.86372996

RESULTS CF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.97003027	0.94583414	-0.96792409	1.00081726
2	0.97002941	-0.94583770	-0.96793108	-1.00081754
3	0.25168247	0.53580191	-1.78711241	1.95425944
4	0.25167543	-0.53579888	-1.78713746	-1.95423795
5	0.00000010	0.26710278	-2.24090416	2.86059552
6	0.00000010	-0.26681661	-2.24551510	-2.86741739

RESULTS CF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.97144662	0.94762010	-0.96904247	1.00080514
2	0.97144593	-0.94762088	-0.96905146	-1.00080547
3	0.25597577	0.54275341	-1.80291116	1.95565286
4	0.25597714	-0.54275353	-1.80292514	-1.95564222
5	0.00000010	0.27664033	-2.30082557	2.86676095
6	0.00000010	-0.27744624	-2.27393152	-2.86450662



RESULTS CF OPTIMIZATION WITH 6 EXTRA DATA PCINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	C.97C74646	0.94593786	-0.96820496	1.00084896
2	0.97C74740	-0.94593893	-0.96820018	-1.00084921
3	0.25317420	0.53401680	-1.78619717	1.95473602
4	0.25317481	-0.53401801	-1.78618621	-1.95473121
5	0.00000010	0.26390985	-2.23469058	2.86348662
6	C.00000010	-0.26397500	-2.23254981	-2.86625171

RESULTS CF OPTIMIZATION WITH 8 EXTRA DATA PCINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	C.97233004	0.94707234	-0.96927693	1.00087271
2	0.97233259	-0.94707302	-0.96923353	-1.00086402
3	0.24416111	0.53457339	-1.78489285	1.95562845
4	C.24415737	-0.53465427	-1.79548721	-1.94953438
5	0.00000010	0.20134605	-1.70725522	2.99741937
6	C.00000010	-0.21428806	-1.61287861	-2.76554177

RESULTS CF OPTIMIZATION WITH 10 EXTRA DATA PCINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	C.97265829	0.94587240	-0.97006813	1.00100830
2	0.97266290	-0.94587709	-0.96753425	1.00089816
3	0.24466698	0.52883260	-1.77855047	1.95554555
4	C.24466804	-0.52886614	-1.78467362	1.95086597
5	C.00000010	0.25592515	-1.69220053	2.78203752
6	C.00000010	-0.15339728	-1.60711291	-3.01800673



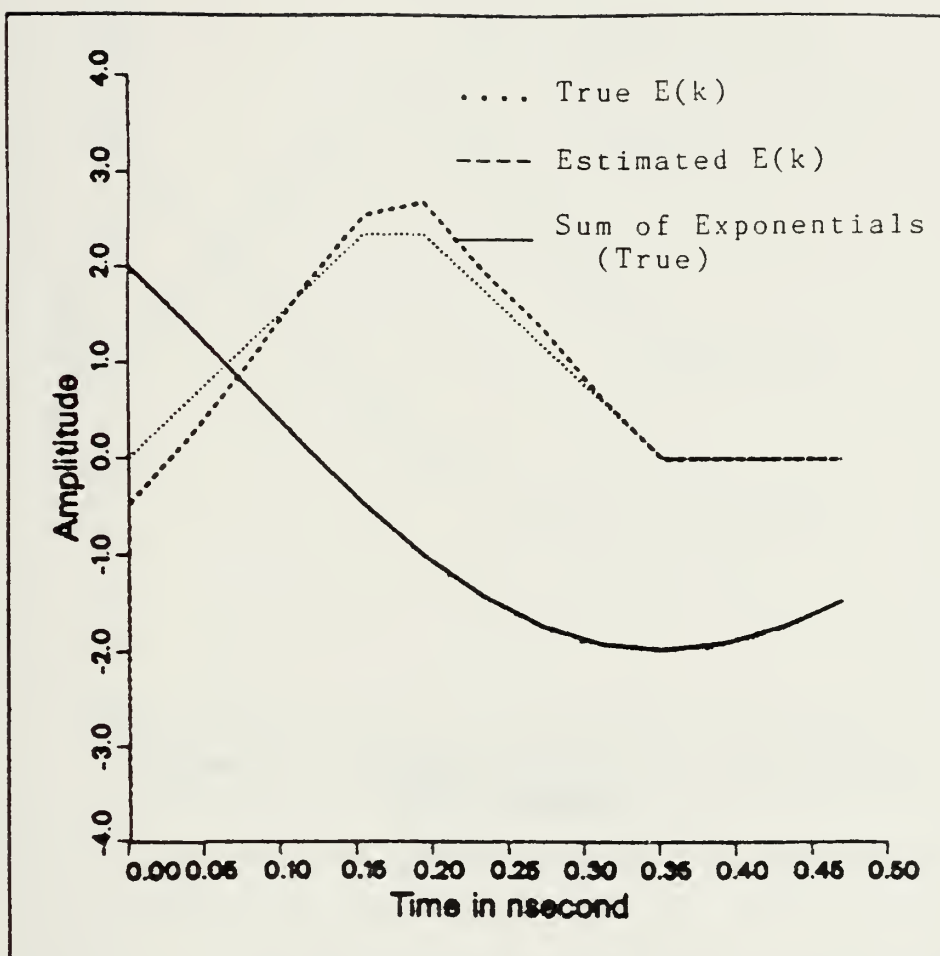


Figure 5.1. Estimated  $E(k)$  for Signal 1 of 15 dB SNR. (32 data points from late time region).



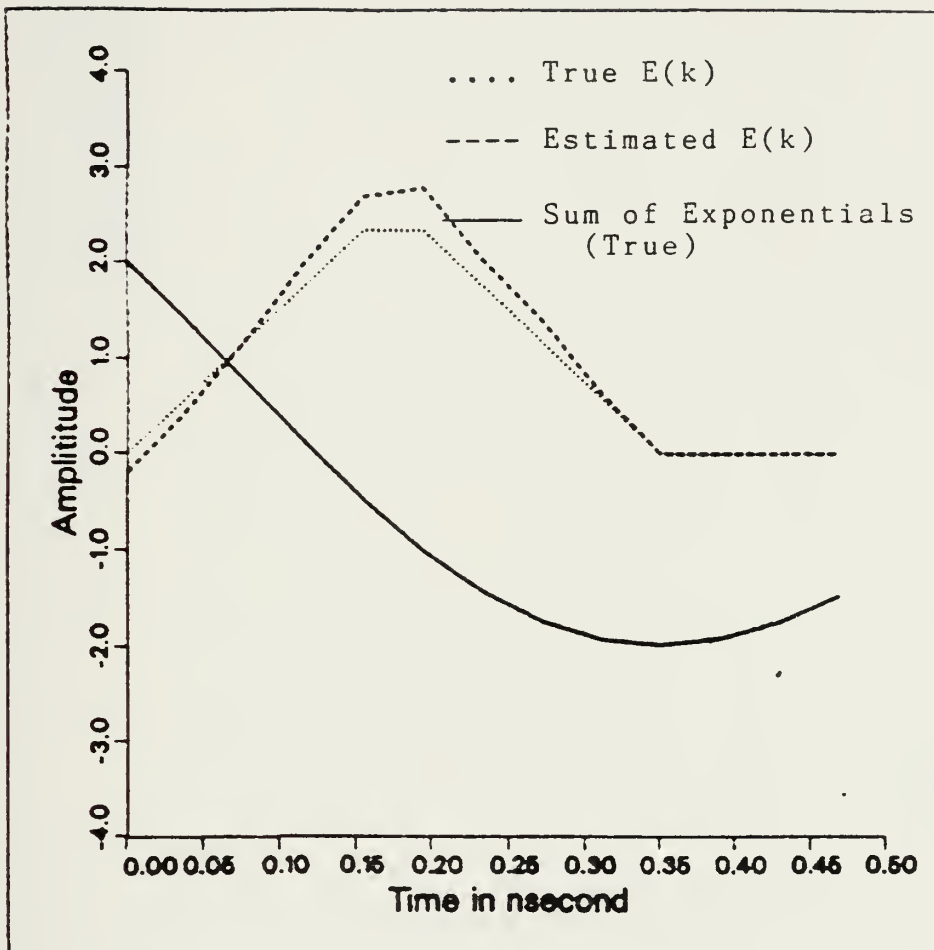


Figure 5.2. Estimated  $E(k)$  for Signal 1 of 15 dB SNR. (48 data points from late time region).





# SIGNAL 1 ( NOISE FREE )

□ : RESIDUE (TRUE)  
 x : POLE (TRUE)

Δ : 1ST STEP (WITH EXTRA 0 POINTS)  
 ° : 2ND STEP (WITH EXTRA 2 POINTS)  
 • : 3RD STEP (WITH EXTRA 4 POINTS)

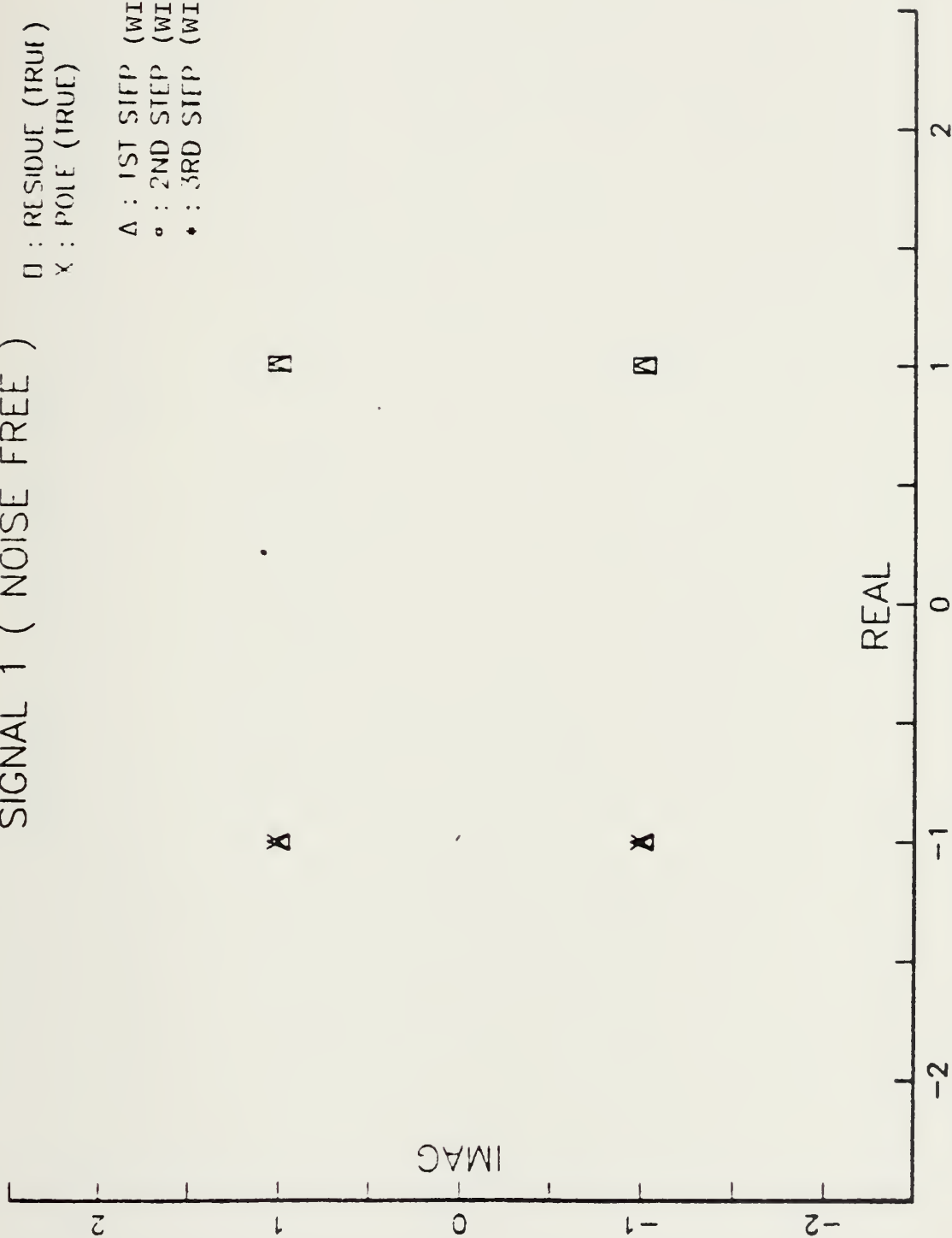


Figure 5.3. Pole and Residue Plot for Signal 1 of Noise Free.



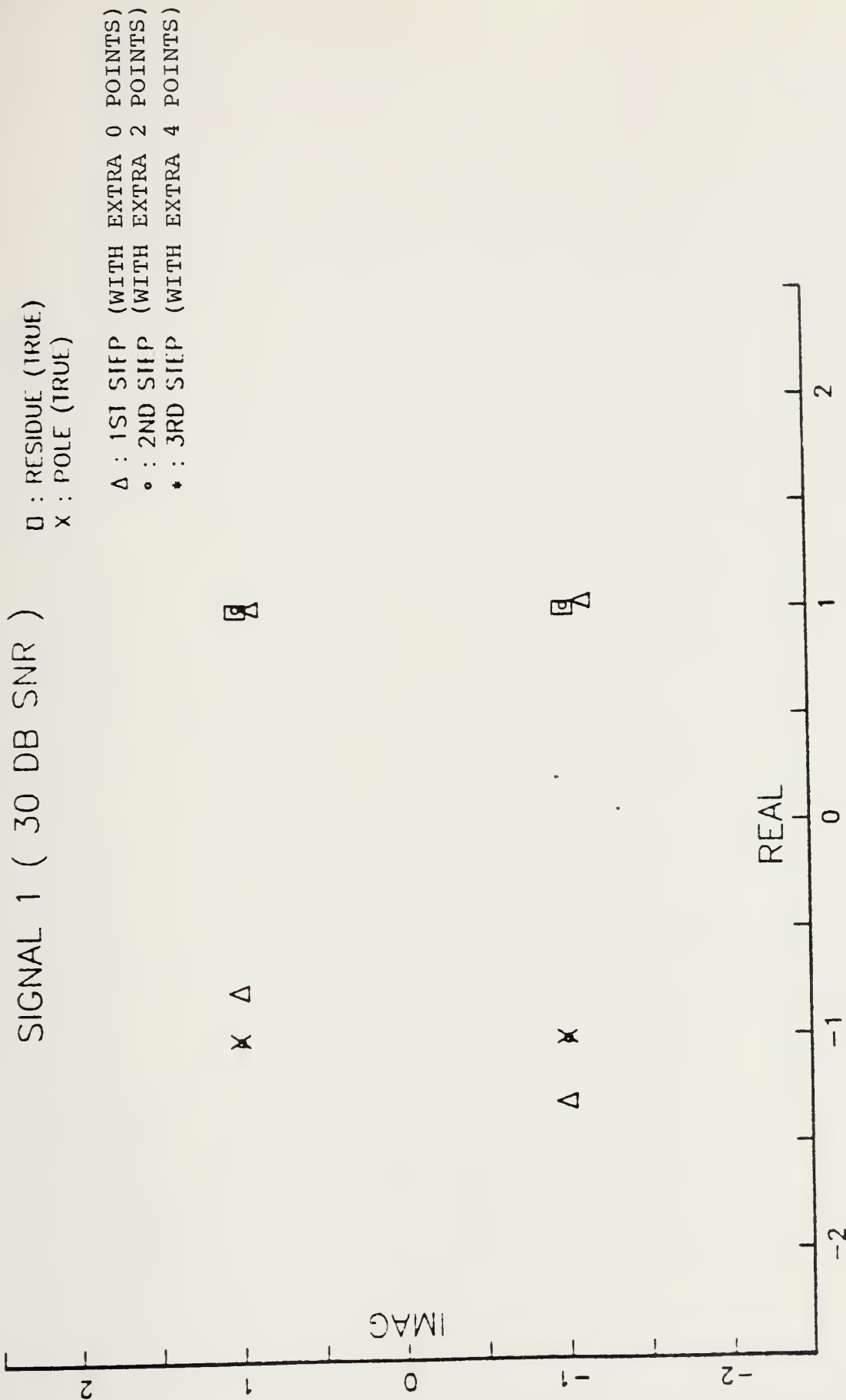


Figure 5.4. Pole and Residue Plot for Signal 1 of 30 dB SNR.



# SIGNAL 1 ( 15 DB SNR )

0 : RESIDUE (IRUF)  
x : POLE (IRUF)

Δ : 1ST SHP (WITH EXTRA 0 POINTS)  
° : 2ND SHP (WITH EXTRA 2 POINTS)  
♦ : 3RD SHP (WITH EXTRA 4 POINTS)

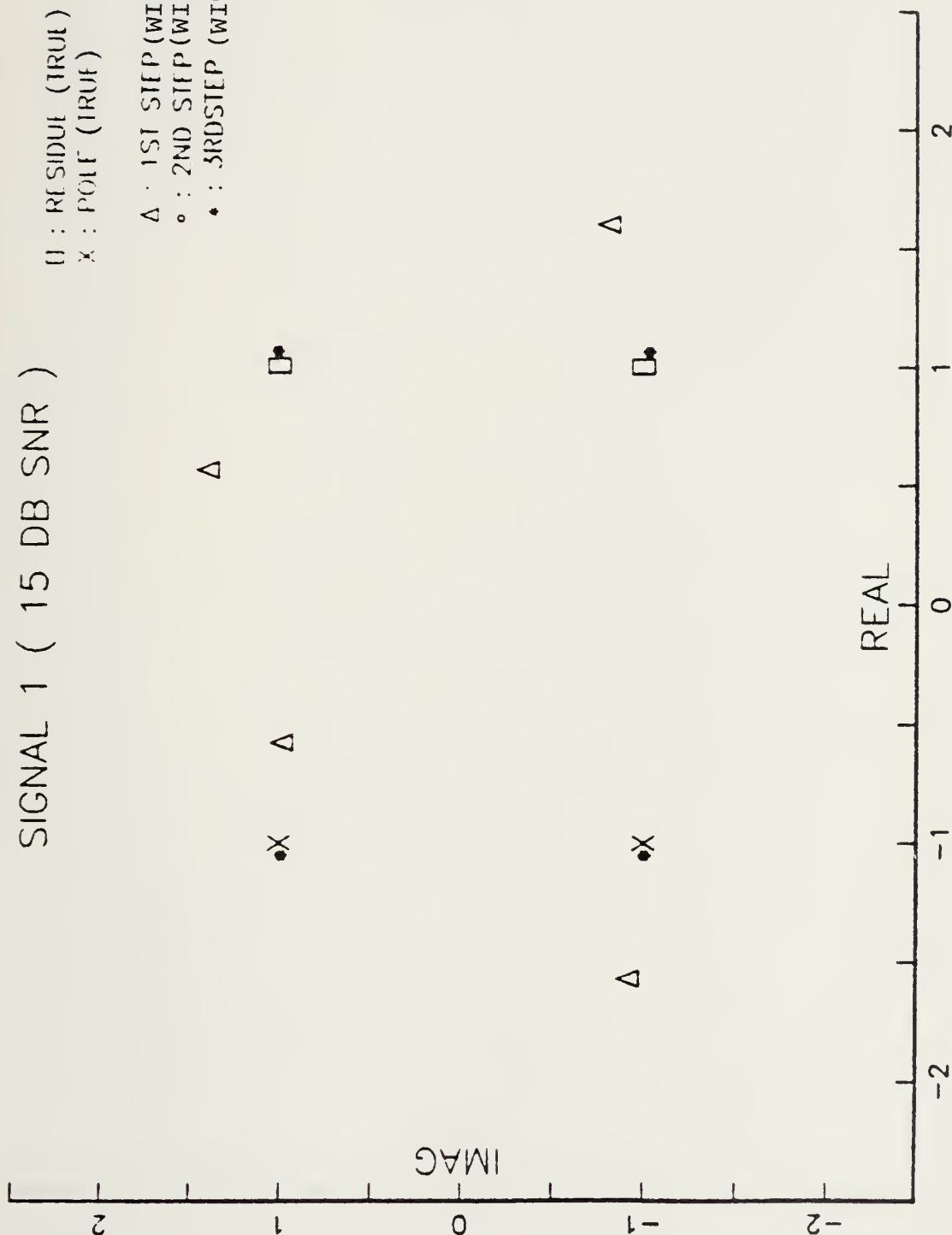


Figure 5.5. Pole and Residue Plot for Signal 1 of 15 dB SNR.



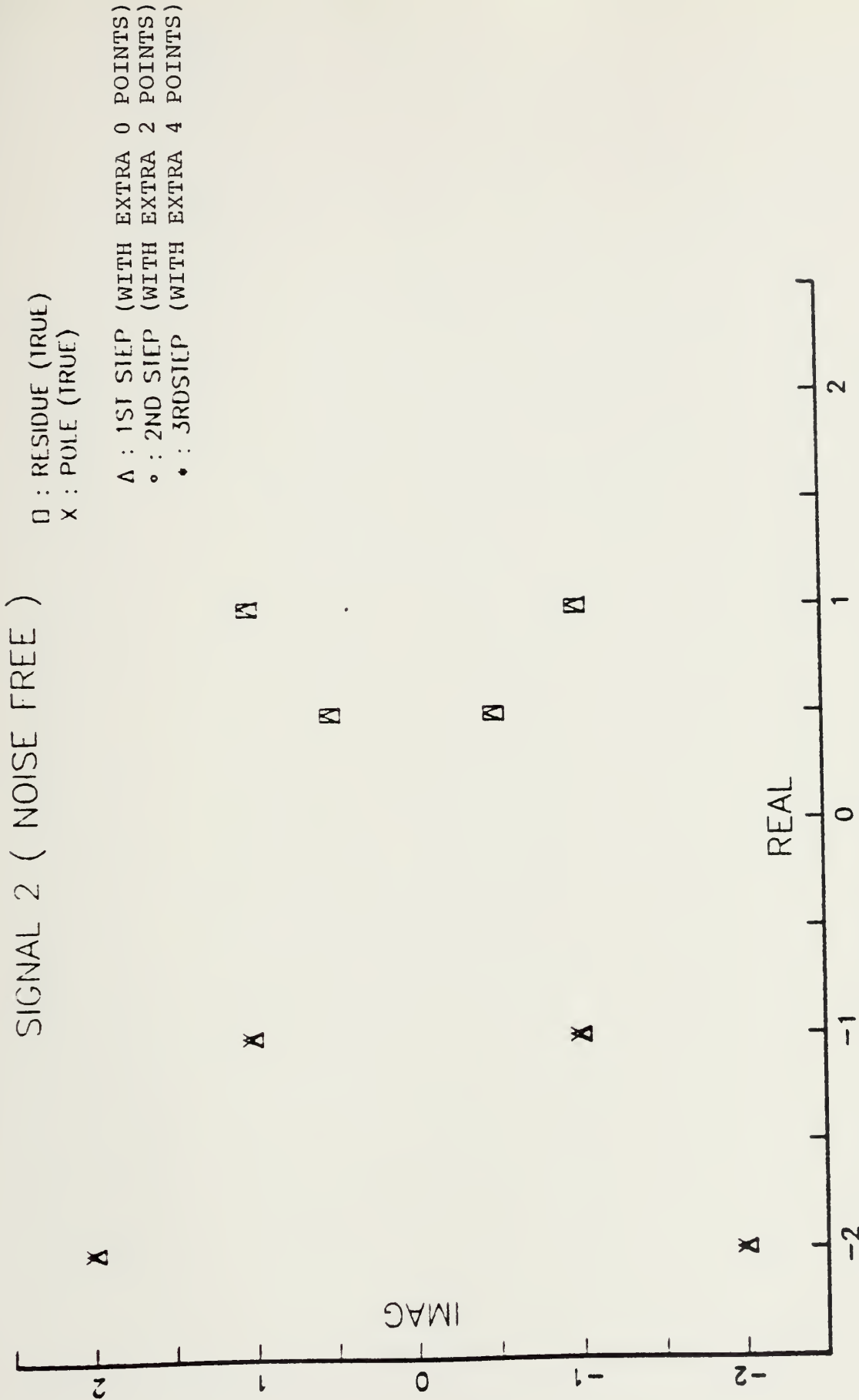


Figure 5.6. Pole and Residue Plot for Signal 2 of Noise Free.





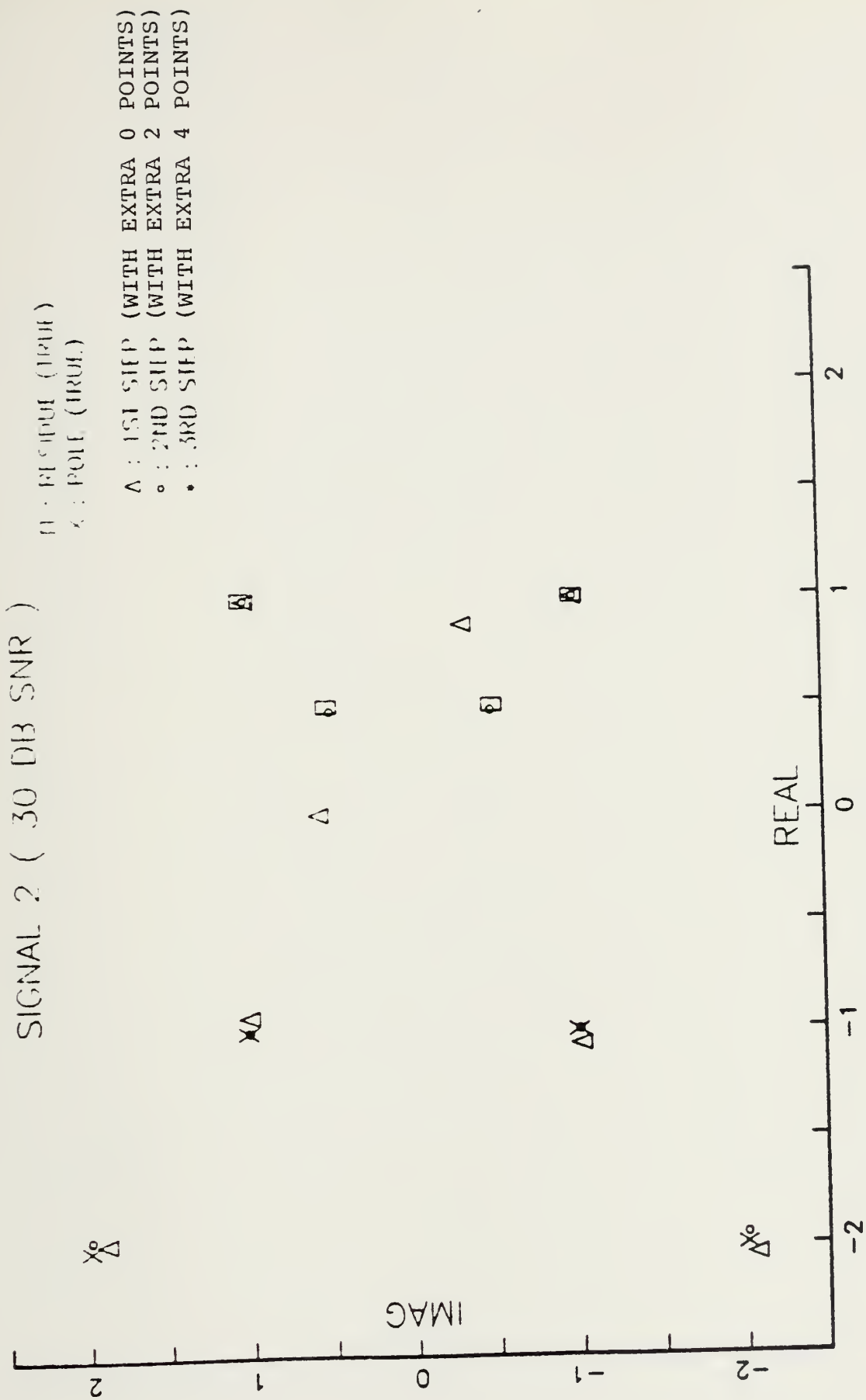


Figure 5.7. Pole and Residue Plot for Signal 2 of 30 dB SNR.



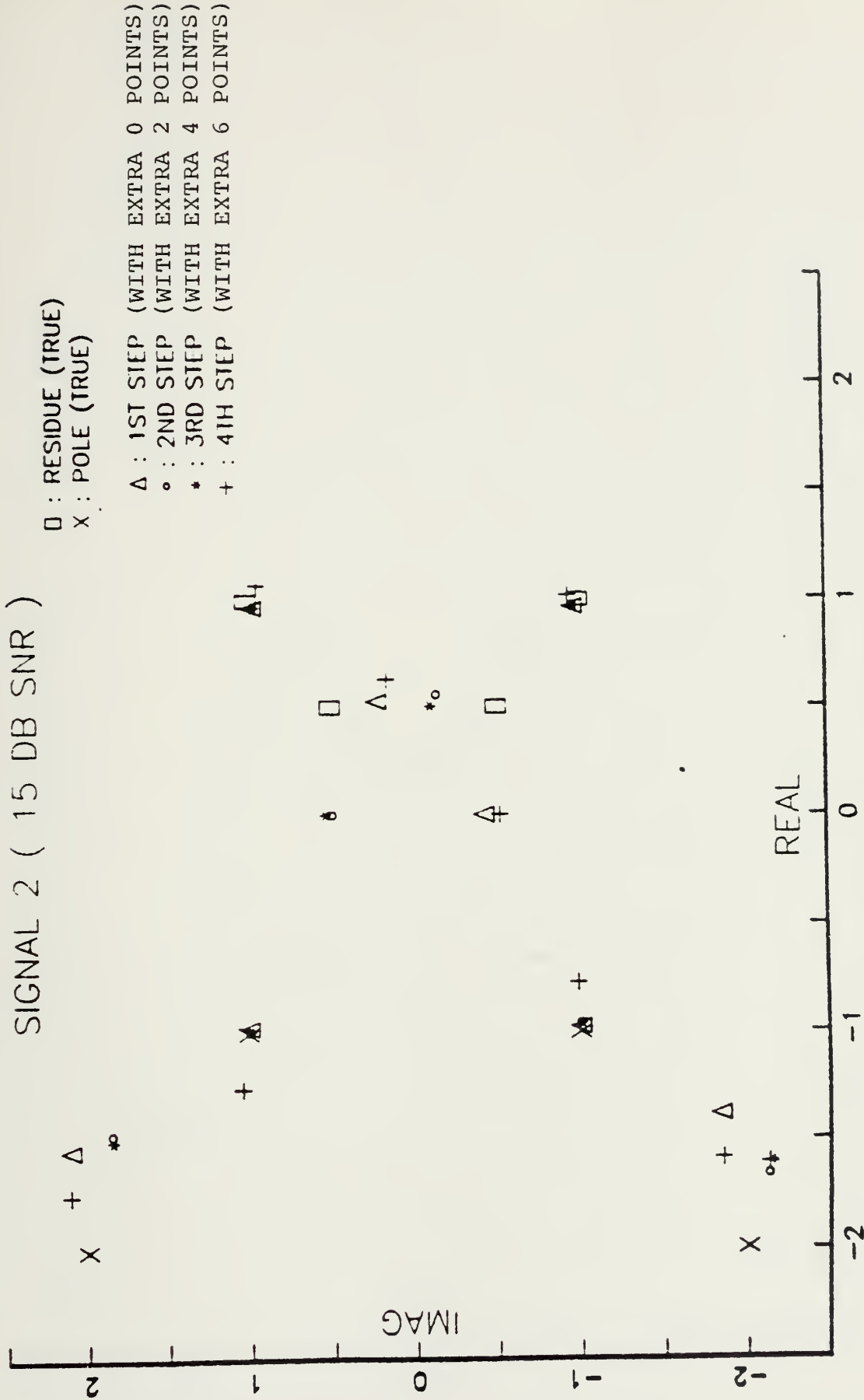


Figure 5.8. Pole and Residue Plot for Signal 2 of 15 dB SNR.



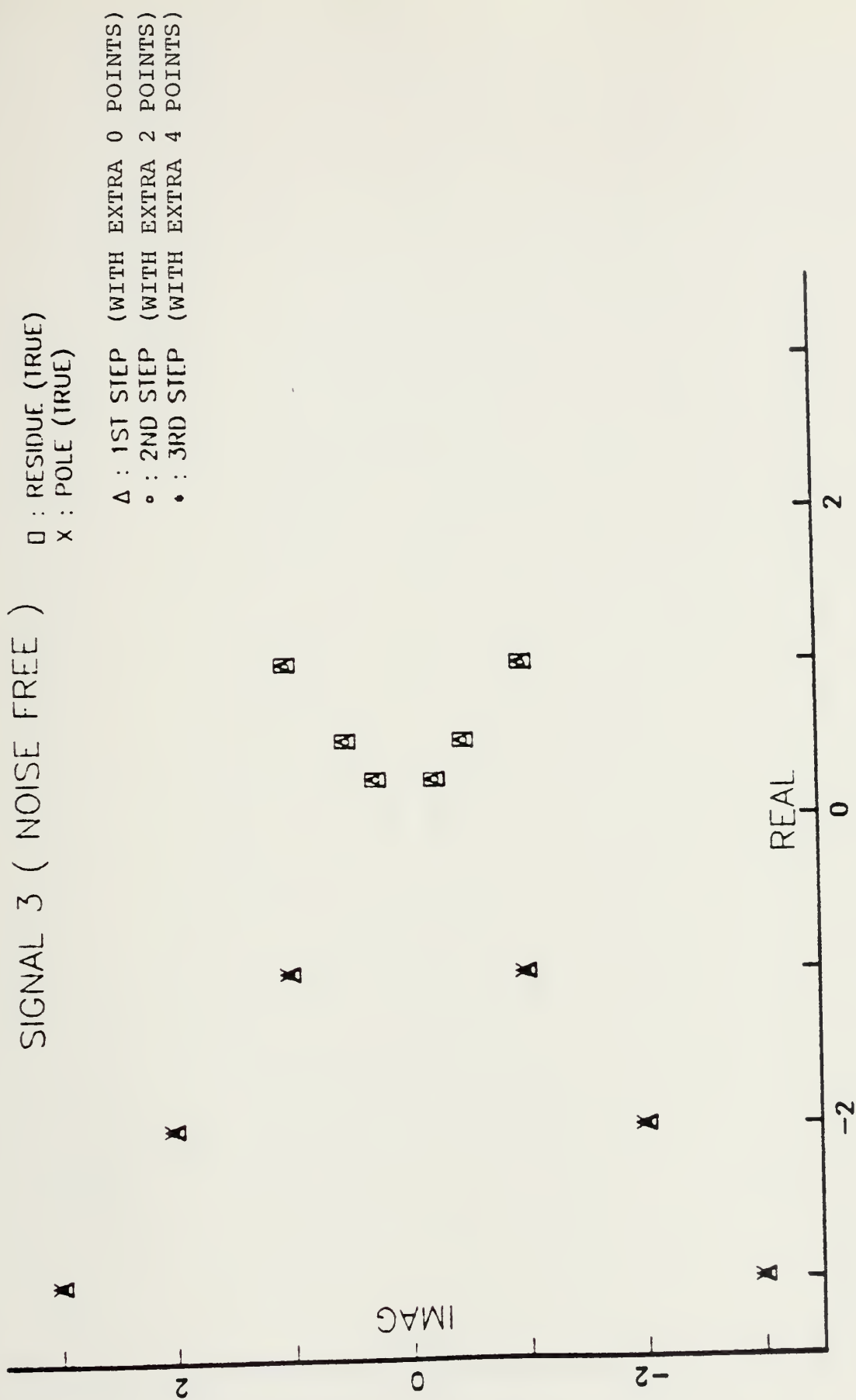


Figure 5.9. Pole and Residue Plot for Signal 3 of Noise Free.



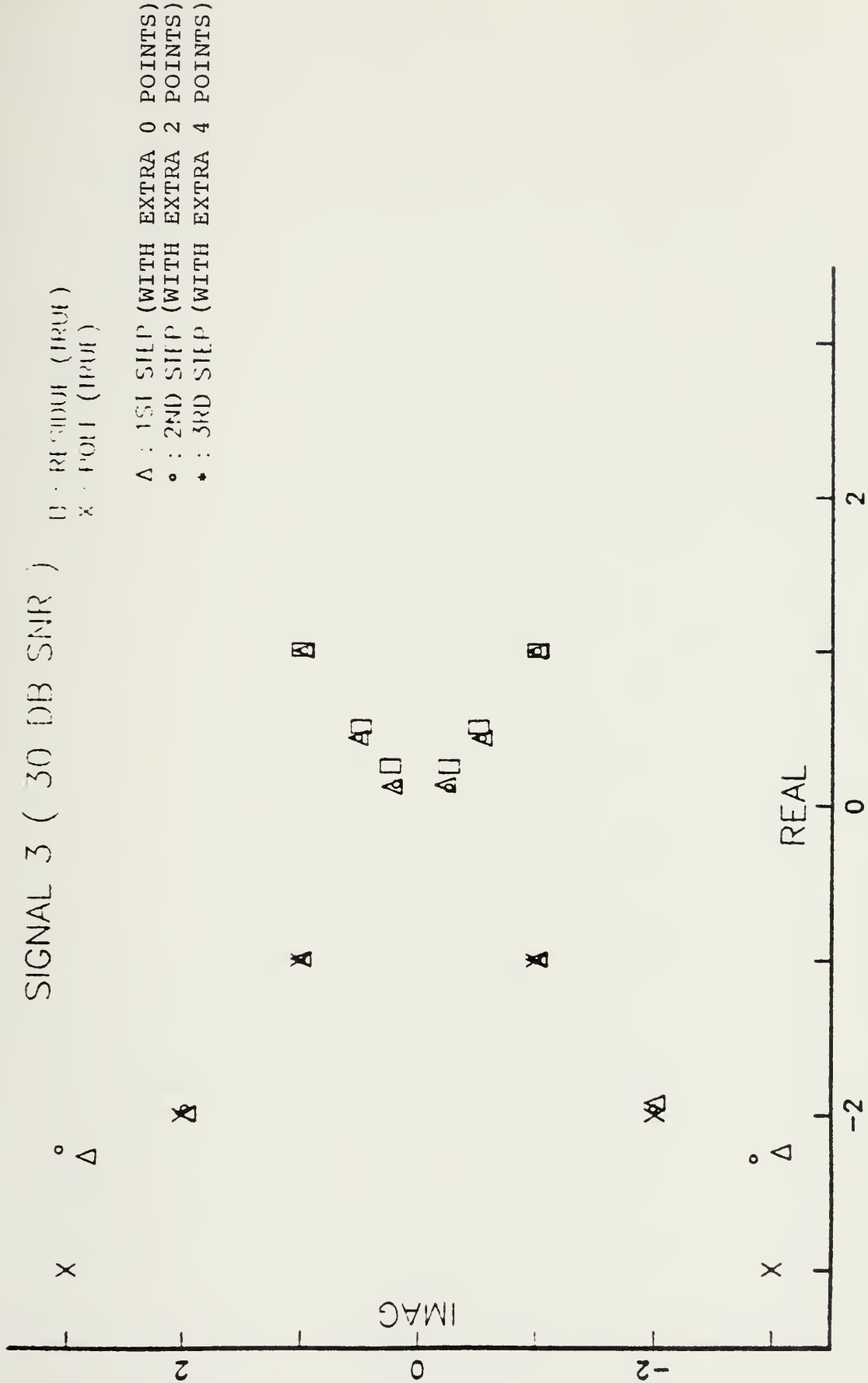


Figure 5.10. Pole and Residue Plot for Signal 3 of 30 dB SNR.





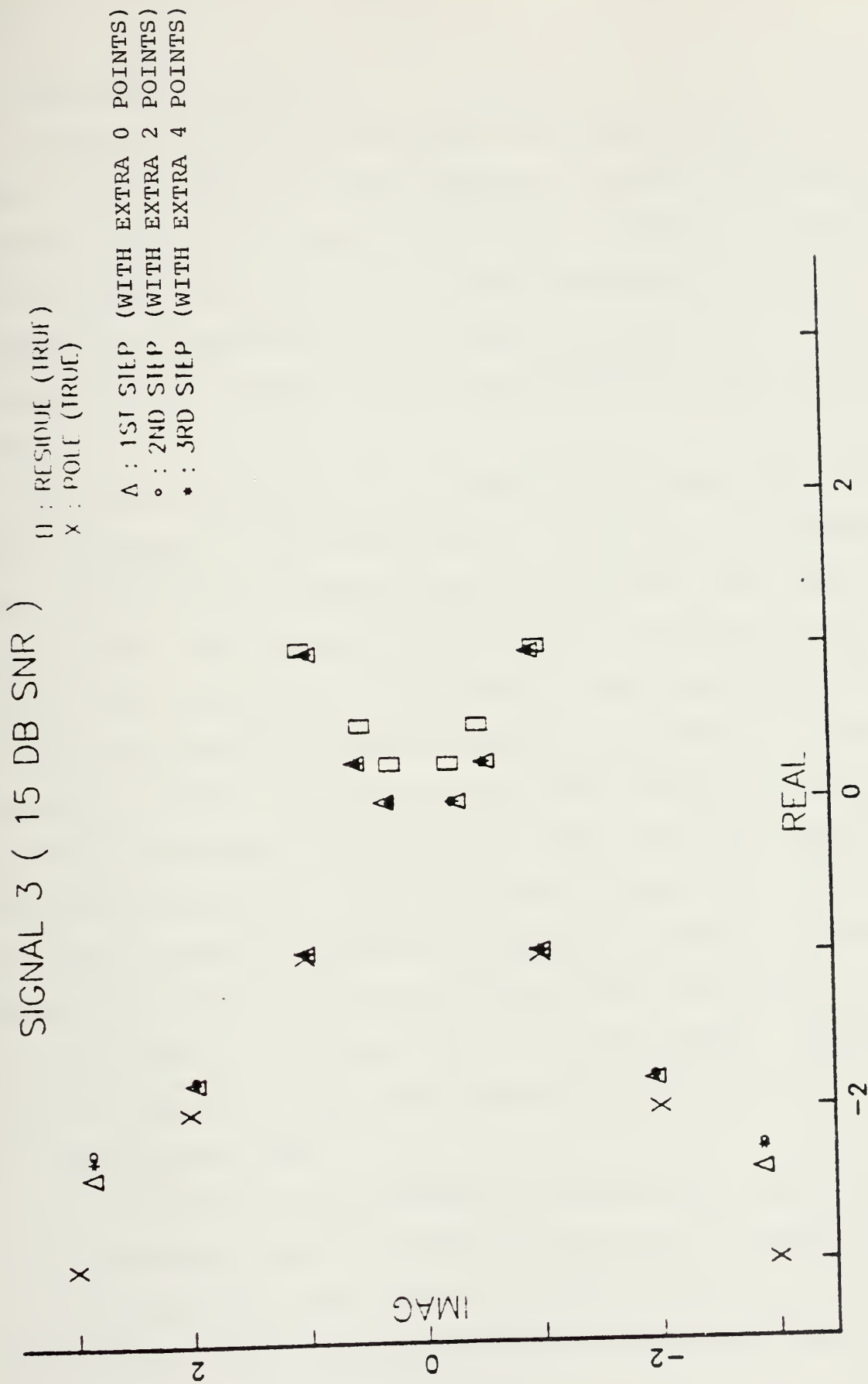


Figure 5.11. Pole and Residue Plot for Signal 3 of 15 dB SNR.



### C. PERFORMANCE EVALUATION

Unlike the traditional methods, this method revealed itself as one that can be applicable to a general scattering problem, including the early time data. A potential superiority of this method over the traditional methods is found in its capability of handling the early time signal that has relatively large energy content and less sensitivity to the fixed noise level.

Regarding the capability of handling the early time signal, the performance of this method seems to be very high compared with the traditional methods which do not model that signal generically. But, two points need to be considered when we ensure this high degree of performance. One important point is that we be able to estimate properly the number of poles to optimally ask for. This will depend upon the SNR of the data, increasing generally with increasing SNR. The actual physical scattering data has an infinite number of poles but, because the energy content of natural modes tends to drop off rapidly beyond the bandwidth of the excitation signal the higher order poles are ignored, blending with the noise present. Another point is that the initial data window has to contain only the late time signal data.

As the results of the tables in the previous pages indicate, converging characteristics of the value of parameters is not blocked by a high level of noise pollution. It has been observed that the computed value of poles are likely to



converge closely to the true values that are listed in Tables I through III either by choosing the number of poles as small or by increasing the size of the data window. Even though it is hard to prove this optimality of this method (due to the non-linearity of the scattering transient response signal model), handling of the early time signal seems to be cast into a simpler problem space and that optimality may be explained by observing the parameters at every processing step. Namely, the parameter estimation of this method is less influenced by the unreliable points than the more accurate ones. Figure 5.1 and 5.2 and the results in the previous section explain the reason why the above assertion is valid. We see that the estimated value of parameters through optimization is not going to worse as we include the early time data points (these data points can be regarded as the unreliable data when we extrapolate the sum of exponentials of the late time region into those of the early time region) into the data window what we see. But, as we are adding more data points of the late time region, improved values of parameters are obtained through processing, as in the results and Fig. 5.2. Although this is true, we still see that the plot of  $E(k)$  in Fig. 5.2 does not provide sufficient data to confirm that an improvement of accuracy depends only on the number of late time data points, because those data points have a lower level of SNR locally. Now, let us get back to Tables IV, VII, and X. For the signal inputs whose SNRs are assumed to be



infinite (noise free), we have a set of parameters whose values are not converging to the true values as we add an increased number of unknowns. These results ensure us that the increased computer round-off errors obtained by more data processing (more parameters to extract) will produce divergence. Even initially with low SNR, the increased signal energy in the additive early-time data may improve the results until, by adding enough extra data points and parameters, round-off error will eventually catch up with the most optimized one and beyond a specific point the results will begin to diverge. So that, in a general sense, we can say that the improvement in accuracy may be obtained either by the number of data points or by making use of the data points of high SNRs. If we consider a signal having high damping coefficients we can not use the signal data what are likely to be hidden by noise. The technique used here in processing the data proceeds by sliding the first data point along the time-axis towards the origin until the data window expands its range up to time-origin by adding the unknown parameters  $E(k)$ 's which is the same as the number of data points in the early time region.

Figures 5.3 through 5.11 shows how rapidly the parameters we want to derive converge to the true values, within a convergence bounds. Here, all the plots were reconstructed from the results.





## VI. SUMMARY

A modified least-squares minimization method has been developed and tested against synthetically generated signal data using a non-linear parameter optimization algorithm. The effect of varying the number of data points either in the region of early time or in the late time under various typical noise environments has been studied, defining the criteria for using this algorithm for the extraction of poles and residues under relatively heavy background noise, as well as for the time varying residue signals modeled by Morgan [Ref. 4].

In an attempt to use this method for signals with high damping coefficients, 3 synthetically generated signals were used as the inputs (Appendix C). Extraction of poles from the direct synthetic signal waveform was done through non-linear numerical evaluations, not by the Prony-based evaluations which were used in the traditional method. The parameter estimating algorithm for this non-linear signal model had worked successfully, providing users with optimized parameters that were converging to the true values, within specified convergence bounds. It was found that although we have the early time signal data having very high SNR, those data can not contribute to improving accuracy by simply increasing the number of extra data points and extra unknown parameters.



In order to get more accurate values of poles, we have to have an increased base of late time signal data, which have relatively low SNR.

It has been shown that this method is "robust" even under heavy noise conditions. But three basic requirements have to be met for using this method as a general methodology; the optimal number of requested poles has to be known in advance to processing, reasonable initial estimation of parameters has to be made, and the transition time from early to late time signal models has to be known a priori.



# APPENDIX A

## SYNTHETIC DATA GENERATION PROGRAM

```

C C C C C C C C
*** DGEN - MAIN PROGRAM : GENERATES SYNTHETIC TRANSIENT RESPONSE ***
*** FOR TRESERACH DATA OF 512 POINTS ***
*** DEC 20, 1983 (#550) CHONG, CHOONG YEON ***
*** IMPLICIT REAL*8 (A-T,C-Z) ***
*** DIMENSION AGDF(500) ***
*** DIMENSION R1(64),R2(64),P1(64),P2(64) ***
*** DIMENSION X0(512),XN0(512) ***
*** DATA IY//Y//,IZ//N, ***
*** DATA D2P31M/2147483647.D0/ ***
*** DATA D2P31/2147483648.D0/ ***
=====
C CLEAR SCREEN
C CALL FRTCMS('CLRSCRN ')
C
C DATA GENERATION MODE SELECTION
61 WRITE(6,61)
  61 FORMAT(1X, '//,10X, '*** SYNTHETIC SIGNAL DATA GENERATION ***',
  10X, 'INTERACTIVE PROGRAM EXECUTION BEGINS',
  10X, '*** THIS PROGRAM GENERATES 512 POINTS OF DIGITAL ***',
  10X, '*** SIGNAL (TRADITIONAL CR NEW MODEL) SYNTHETICALLY ***',
  10X, '*** EITHER IN NOISE-FREE CR NOISE POLLUTING MODE ***',
  10X, '*** USER IS ASKED TO INPUT THE DATA GENERATING PARAMETERS ***',
  10X, 'INTERACTIVELY, //,1X, 'TYPE <G> TO GC IF YOU READY TO INPUT',
  * READ(5,1) IANS
  * REFORMAT(1)
51 FORMAT(1X)
C INITIALIZE IGN. MAXIMUM 64 POLES
DO 10 I=1,64
  R1(I)=0
  R2(I)=0
  P1(I)=0
  P2(I)=0
10
C GET THE DATA IDENTIFIERS
  CALL HEADER(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2)
C GET THE DATA PARAMETERS
  CALL PAFA(N1,T2,S1)
C GET THE DATA PARAMETERS-S-POLES/RESIDUES INTERACTIVELY
  WRITE(6,311)

```



```

311 FORMAT(IX,'DO YOU WANT TO GET TRUE POLE/ZERCS FROM THE PRE-DEFINED
* DATA FILE? -<Y/N>'),
READ(5,51) IANS
IF(IANS.NE.IV) GO TO 312
CALL DATA(N1,R1,R2,P1,P2)
GO TO 313

C 312 CALL RPIN(N1,R1,R2,P1,P2)
C
C VERIFICATION POINT - DISPLAY OF POLE AND RESIDUE TABLE
312 WRITE(6,62)
62 FORMAT(IX,'DO YOU WANT TO HAVE THE RES/POLES IN A TABLE FORM? - <Y
*/N>'),
READ(5,51) IANS
IF(IANS.NE.IV) GO TO 990
CALL FRICMS('CLRSCRN')
CALL DISPI(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N1,R1,R2,P1,P2)
990 CONTINUE

C NOISE-FREE SYNTHETIC DATA GENERATOR
WRITE(6,63)
63 FORMAT(IX,'DO YOU WANT TO GENERATE A NCISE-FREE SIGNAL? - <Y/N>'),
READ(5,51) IANS
IF(IANS.NE.IV) GO TO 991
CALL GNFD(N1,R1,R2,P1,P2,T2,S1,M9,X0)
991 CONTINUE

C
C CALL HEADER(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2)
C
C NOISE-FREE DATA SIGNAL DISPLAY
WRITE(6,58)
58 FORMAT(IX,'DO YOU WANT TO SEE THE NCISE FREE SIGNAL IN A TABLE FOR
*M? - <Y/N>'),
READ(5,51) IANS
IF(IANS.NE.IV) GO TO 992
N2=0
CALL DISP2(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N2,T2,S1,X0)
992 CONTINUE
IFLAG=0
GO TO 5111

C NOISE POLLUTED SYNTHETIC DATA GENERATOR
6111 IFLAG=1
WRITE(6,59)
59 FORMAT(IX,'DO YOU WANT TO GENERATE A NCISE-POLLUTED SIGNAL DATA? -
* <Y/N>'),
READ(5,51) IANS
IF(IANS.EQ.IZ) GO TO 9999

```





```

C GENERATE THE GAUSSIAN DISTRIBUTION
C CALL GALSS(AGDF,IWA,IGA)
C GO TO 444
DGE00C970
DGE00C980
DGE00C990
DGE001C00
DGE001C10
DGE001C20
DGE001C30
DGE001C40
DGE001C50
DGE001C60
DGE001C70
DGE001C80
DGE001C90
DGE001100
DGE001110
DGE001120
DGE001130
DGE001140
DGE001150
DGE001160
DGE001170
DGE001180
DGE001190
DGE001200
DGE001210
DGE001220
DGE001230
DGE001240
DGE001250
DGE001260
DGE001270
DGE001280
DGE001290
DGE001300
DGE001310
DGE001320
DGE001330
DGE001340
DGE001350
DGE001360
DGE001370
DGE001380
DGE001390
DGE001400
DGE001410
DGE001420
DGE001430
DGE001440

C OPTIONS FOR POLLUTING MODE
444 WRITE(6,64)
64 FORMAT(IX,'CHOOSE ONE OF THE THREE MODES OF NOISE POLLUTION',
*//,
IX,'1: IN TERMS OF SNR(AVERAGE) IN DECIBEL',
*//,
IX,'2: IN TERMS OF SNR(PEAK) IN DECIBEL',
*//,
READ(5,*) IR
C ADJUST NOISE LEVEL IN QUICK
C CALL TRAP(X0,Q,Q1)
C
C WRITE(6,641) Q
641 FORMAT(IX,'AVERAGE SIGNAL POWER : ',F12.6)
IF(IR.EQ.2) GO TO 222
IF(IR.GT.2) GO TO 444
C
C WRITE(6,66)
66 FORMAT(IX,'ENTER SNR(AVERAGE) IN DECIBEL')
READ(5,*) W1
W3=10.*W1/10.)
DEV=DSCRT(Q/W3)
GOTO 4111
C
C WRITE(6,67)
67 FORMAT(IX,'ENTER SNR(PEAK) IN DECIBEL')
READ(5,*) W2
W3=10.*W2/10.)
DEV=DSCRT(Q1/W3)
C
C 4111 CONTINUE
C SEED OF RANDOM NOISE GENERATOR WITH UNIFORM DISTRIBUTION
DSEED=12457.D0
C
C DO NOISE AVERAGING AND ADJUST IF NECESSARY
WRITE(6,68)
68 FORMAT(IX,'ENTER THE NUMBER OF NOISE AVERAGINGS - RECOMMEND:100')
READ(5,*) N2
DO 20 I=1,N2
DO 20 J=1,512
I*J=0
RANDOM NUMBER GENERATION
CSEED = DMOD(16807.D0*DSEED,D2P31M)
RV = DSEED / D2P31

```



```

C          GET GCF INDEPENDENT VARIABLE
RN=RV
IF(RN.LE.0.5) GOTO 555
RN=1-RN
IS1=1
C          CALL TRANF(AGCF,IWA,DEV,RN,X)
C          IF(IS1.EQ.0) GOTO 666
XS=-X
GOTO 777
XS=X
XNO(J)=XNO(J)+X9
201 CONTINUE
20 CONTINUE
C          CALL NTEST(IR,W1,W2,Q,Q1,XNO,NFLAG,N2)
C          ADC NOISE
DO 30 I=1,512
30 X0(I)=X0(I)+XNO(I)
C          CHANGE IF NECESSARY
CALL HEADER(NT1,NT2,NW1,NW2,ND1,ND2,NF1,ND2)
C          NOISE POLLUTED SIGNAL DISPLAY
WRITE(6,65)
65 FORMAT(1X,'DO YOU WANT TO HAVE THE NOISE POLLUTED DATA IN A TABLE
*FCRM? - <Y/N>')
READ(5,51)IANS
IF(IANS.EQ.12) GOTO 9999
C          CALL DISP2(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N2,S1,X0)
C          SYNTHETIC DATA STORAGE
5111 WRITE(6,71)
71 FORMAT(1X,'ENTER THE FILE NUMBER OF A DATA FILE INTO WHERE YOU WANT
*1 TO STORE THE DATA,/,1X,'IF YOU DO NOT WANT TO, ENTER 13,/,
*,1X,1 : HIGH-DAMPING/NOISE-FREE/CONSTANT RESIDUE DATA FILE',
*,1X,2 : /30DB-SNR /TIME VARYING RESIDUE DATA FILE',
*,1X,3 : /15DB-SNR /TIME VARYING RESIDUE DATA FILE',
*,1X,4 : /30DB-SNR /TIME VARYING RESIDUE DATA FILE',
*,1X,5-6 : NO NOT USE,
*,1X,7 : LCW-DAMPING/NOISE-FREE/CONSTANT RESIDUE DATA FILE',
*,1X,8 : /30DB-SNR /TIME VARYING RESIDUE DATA FILE',
*,1X,9 : /15DB-SNR /TIME VARYING RESIDUE DATA FILE',
*,1X,10 : /30DB-SNR /TIME VARYING RESIDUE DATA FILE',
*,1X,11 : /15DB-SNR /TIME VARYING RESIDUE DATA FILE')

```



```

C      READ(5,*) IFILE      GO TO 7
C      IF(IFILE.EQ.13)      GO TO 7
C      WRITE(IFILE,8) NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N2,I2,S1
C      8  FORMAT(2X,2A4,/,2X,2A4,/,2X,2A4,/,2X,F14.8)
C      *  FORMAT(2X,2A4,/,2X,14,/,2X,F14.8,/,2X,F14.8)
C      DO 7 I=1,512,4
C      WRITE(IFILE,9) X0(I),X0(I+1),X0(I+2),X0(I+3)
C      9  FORMAT(2X,4(2X,F14.8))
C      7  CONTINUE
C      IF(IFLAG.EQ.0) GO TO 6111
C      995 CONTINUE
C      111 CONTINUE
C      999$ CONTINUE
C      CALL FRICMS('CLRSCRN ')
C      72  WRITE(6,72)
C      FORMAT(1X,////,10X,*** PROCESSING COMPLETED ***)
C      STCP
C      END
C      =====
C      = SUBROUTINE HEADER - CAN BE USED WHENEVER USER NEEDS TO CHANGE =
C      =====
C      SUBROUTINE HEADER(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2)
C      IMPLICIT REAL*8(A-H,O-Z)
C      DATA IV,'Y',IZ,'N' /
C      CALL FRICMS('CLRSCRN ')
C      55  WRITE(6,60)
C      60  FORMAT(1X,CO YOU WANT TO INITIALIZE/CHANGE THE HEADER? - <Y/N>.)
C      READ(5,10) IANS
C      IF(IANS.EQ.12) GO TO 10
C      IF(IANS.NE.IV) GO TO 59
C      61  WRITE(6,61)
C      61  FORMAT(1X,ENTER THE TARGET TYPE WITHIN 8 CHARS,
C      *,1X,EX: TGT-1.)
C      READ(5,51) NT1,NT2
C      WRITE(6,62)
C      62  FORMAT(1X,ENTER THE WAVEFORM TYPE WITHIN 8 CHARS,

```



```

* /, 1X, .EX : PDIN - HIGH DAMPING/TIME VARYING RESIDUE/NOISE ADDED.,
* /, 1X, . : LDCF - LOW DAMPING/CCNSTANT RESIDUE/NOISE FREE.)
      REAC(5,51) NW1,NW2
      WRITE(6,63)
63  FORMAT(1X, 'ENTER THE DATE WITHIN 8 DIGITS.',
* /, 1X, .EX : DEC. 20.)
      REAC(5,51) ND1,ND2
      WRITE(6,64)
64  FORMAT(1X, 'ENTER THE NAME OF FILE WITHIN 8 CHARS.',
* /, 1X, .EX : FILECOL.)
      REAC(5,51) NF1,NF2
5C  FORMAT(1A1)
51  FORMAT(2A4)
      1C CONTINUE
      RETURN
      END
=====
= SUBROUTINE PARA - INITIALIZE THE NUMBER OF POLES, SIZE OF TIME =
= WINDOW AND VERTICAL SCALE IN GRAPHIC =
=====
      SUBROUTINE PARA(N1,T2,S1)
      IMPLICIT REAL*8(A-H,O-Z)
      DATA IB/,B/,IG/,G/
      1 CALL FRTCMS('CLRSCRN ')
      60  WRITE(6,60)
      FORMAT(1X, 'ENTER THE NUMBER OF POLES : MAX 64')
      REAC(5,51) N1
      61  WRITE(6,61)
      FORMAT(1X, 'ENTER THE SIZE OF TIME WINDOW : 20 NANO SECONDR RIGID.')
      REAC(5,51) T2
      62  WRITE(6,62)
      FORMAT(1X, 'ENTER A VERTICAL SCALE FACTOR : 1 INITIALLY.')
      REAC(5,51) S1
      2  WRITE(6,69)
      65  FORMAT(1X, 'TYPE <B> TO CHANGE THE ABOVE INITIALIZATIONS.',
* /, 1X, 'OTHERWISE, TYPE <G> TO GO ON.')
      REAC(5,51) IANS
      51  IF(IANS.EQ.IB) GO TO 1

```





```

C      IF (IANS.NE.IG) GC TC 2
C      RETURN
C      END
C
C      =====
C      = SUBROUTINE RPIN - INPUT THE DATA GENERATING PARAMETER
C      =====
C      SUBROUTINE RPIN(ISN1,SRL,SR2,SP1,SP2)
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION SRL(64),SR2(64),SP1(64),SP2(64)
C      CALL FRTCMS('CLRSCRN ')
C      WRITE(6,60)
C      60 FORMAT(IX,' YOU ARE ASKED TO INPUT INTERACTIVELY THE VALUE OF PARAM
C      *ETERS.')
C      DO 10 I=1,ISN1
C      10 WRITE(6,61) I, I, I, I
C      61 FORMAT(IX,' R-RE(,I2,)=,2X,' R-IM(,I2,)=,')
C      1C READ(5,*) SRL(I),SR2(I),SP1(I),SP2(I)
C      RETURN
C      END
C
C      =====
C      = SUBROUTINE DISP1 - TABULATES THE STATUS OF PARAMETERS
C      =====
C      SUBROUTINE DISP1(NSI1,NSI2,NSW1,NSW2,NSD1,NSD2,NSF1,NSF2,
C      * NSN1,SRL,SR2,SP1,SP2)
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION SRL(64),SR2(64),SP1(64),SP2(64)
C      1 CALL FRTCMS('CLRSCRN ')
C      WRITE(6,211)
C      311 FORMAT(IX,' ENTER THE FILE LOCATION INTO WHERE THE POLE AND RESIDUE
C      * DATA ARE STORED.',
C      */,IX,' 11 : RESERVED FOR HIGH DAMPING DATA',
C      */,IX,' 12 : RESERVED FOR LOW DAMPING DATA',/)
C      READ(5,*) IFILE

```



```

C      WRITE(IFILE,59) NST1,NST2,NSW1,NSW2,NSD1,NSD2,NSF1,NSF2,NSN1
C      WRITE(6,59) NST1,NST2,NSW1,NSW2,NSD1,NSD2,NSF1,NSF2,NSN1
59  FORMAT(2X, 'TARGET TYPE:',2A4,
//, 2X, 'WAVEFORM TYPE:',2A4,
//, 2X, 'CONTACT DATE:',2A4,
//, 2X, 'FILE NUMB:',2A4,
//, 2X, 'NUMB. OF POLE:',14,/)

C      WRITE(IFILE,60)
C
C      WRITE(6,60)
60  FORMAT(2X,15X, 'TABLE OF RESIDUES AND POLES',/,
//, 2X,15X, '=====',/,
//, 2X, 'PAIR #',4X, 'RES-REAL',4X, 'RES-IMAG',/,
//, 2X, 'POLE-REAL',4X, 'POLE-IMAG',)
C      DO 10 I=1,NSN1
C
C          WRITE(IFILE,61) I,SR1(I),SR2(I),SP1(I),SP2(I)
C
C          WRITE(6,61) I,SR1(I),SR2(I),SP1(I),SP2(I)
61  FORMAT(2X,14,3X,4(F12.8,1X))
C      CONTINUE
C      RETURN
C      END
C
C      =====
C      = SUBROUTINE GNFD - GENERATES NOISE FREE SIGNAL
C      =
C      = 6 OPTIONS FOR RESIDUE FUNCTION SELECTION
C      =
C      =====
C      SUBROUTINE GNFD(NN1,SR1,SR2,SP1,SP2,ST2,SS1,M9,SXJ)
C
C      IMPLICIT REAL*8(A-F,H,O-Z)
C      DIMENSION SR1(64),SR2(64),SP1(64),SP2(64),SX0(512),POWER(10)
C      DATA IG,'G',IB,'B',/
C
C      PI=3.141592654
C
C      1 CALL FRICMS('CLRSCRN ')
C
C      111 WRITE(6,60)
60  FORMAT(1X, 'CHOOSE ONE OF THE FOLLOWING ENTIRE FUNCTION',
//, 1X, 'CPTICN 4 IS RECOMMENCED RIGOROUSLY',
//, 2X,5X, '1 : DC',
//, 2X,5X, '2 : POSITIVE LINEAR',
//,
DGE03370
DGE03380
DGE03390
DGE03400
DGE03410
DGE03420
DGE03430
DGE03440
DGE03450
DGE03460
DGE03470
DGE03480
DGE03490
DGE03500
DGE03510
DGE03520
DGE03530
DGE03540
DGE03550
DGE03560
DGE03570
DGE03580
DGE03590
DGE03600
DGE03610
DGE03620
DGE03630
DGE03640
DGE03650
DGE03660
DGE03670
DGE03680
DGE03690
DGE03700
DGE03710
DGE03720
DGE03730
DGE03740
DGE03750
DGE03760
DGE03770
DGE03780
DGE03790
DGE03800
DGE03810
DGE03820
DGE03830
DGE03840

```



```

C
*//,
2X,5X,.3 : NEGATIVE LINEAR.,
*//,
2X,5X,.4 : TRAPEZOIDAL.,
*//,
2X,5X,.5 : CCNSTANT.,
*//,
2X,5X,.6 : TRIANGULAR.)
SS,1=1.
READ(5,*) M9
IF (M9.GT.6) GOTC 111
IF (M9.EC.5) GOTC 999
C
WRITE(6,61)
61 FORMAT(1X, SPECIFY THE POINT AT WHICH E(K) IS TO BE SET TO ZERO.,
*//,
1X, RECOMMENDED 10 : WHEN E(K) : A TRAPEZOIDAL.)
READ(5,*) M3
GO TO 888
C
999 M3=1
888 TO=ST2/511.
C
T9=0.
DO 10 I=1,M3
X9=0.
X8=C.
DO 20 J=1,MN1
X6=SR1(J)*DCOS(2*PI*SP2(J)*T9)-SR2(J)*DSIN(2*PI*SP2(J)*T9)
X5=X9+X8*DEXP(SPI(J)*T9)
20 X0(I)=X5.
T9=T9+T0
POWER(I)=X8*2
10 CONTINUE
TP=0
DO 40 I=1,M2
TP=TP+POWER(I)**2
40
C
4 WRITE(6,73)
73 FORMAT(1X, TYPE <B> IF YOU WANT TO CHOOSE ANOTHER OPTION.,
*//,
1X, OTHERWISE TYPE <G> TO GC.,
READ(5,*) IANS
IF (IANS.EQ.1) GO TC 1
IF (IANS.NE.1) GO TO 4
C
CALL FRTCMS('CLRSCRN ')
C
IF (M9.EC.5) GO TO 99
IF (M9.EC.1) GO TO 88
IF (M9.EC.2) GO TO 77
IF (M9.EC.3) GO TO 66
IF (M9.EC.4) GO TO 55
C

```



```

C TRIANGULAR FUNCTION OF E(K)
  XTR6=DS CRT(TP/6.)
  SX0(2)=SX0(2) + XTR6
  SX0(3)=SX0(3) + 2*XTR6
  SX0(4)=SX0(4) + XTR6
  GO TO 55
DGE04330
DGE04340
DGE04350
DGE04360
DGE04370
DGE04380
DGE04390
DGE04400
DGE04410
DGE04420
DGE04430
DGE04440
DGE04450
DGE04460
DGE04470
DGE04480
DGE04490
DGE04500
DGE04510
DGE04520
DGE04530
DGE04540
DGE04550
DGE04560
DGE04570
DGE04580
DGE04590
DGE04600
DGE04610
DGE04620
DGE04630
DGE04640
DGE04650
DGE04660
DGE04670
DGE04680
DGE04690
DGE04700
DGE04710
DGE04720
DGE04730
DGE04740
DGE04750
DGE04760
DGE04770
DGE04780
DGE04790
DGE04800

C TRAPEZOIDAL FUNCTION OF E(K)
55 XTR4=DS CRT(TP/60.)
  SX0(2)=SX0(2) + XTR4
  SX0(3)=SX0(3) + 2*XTR4
  SX0(4)=SX0(4) + 3*XTR4
  SX0(5)=SX0(5) + 4*XTR4
  SX0(6)=SX0(6) + 4*XTR4
  SX0(7)=SX0(7) + 3*XTR4
  SX0(8)=SX0(8) + 2*XTR4
  SX0(9)=SX0(9) + XTR4
  GO TO 55
DGE04400
DGE04410
DGE04420
DGE04430
DGE04440
DGE04450
DGE04460
DGE04470
DGE04480
DGE04490
DGE04500
DGE04510
DGE04520
DGE04530
DGE04540
DGE04550
DGE04560
DGE04570
DGE04580
DGE04590
DGE04600
DGE04610
DGE04620
DGE04630
DGE04640
DGE04650
DGE04660
DGE04670
DGE04680
DGE04690
DGE04700
DGE04710
DGE04720
DGE04730
DGE04740
DGE04750
DGE04760
DGE04770
DGE04780
DGE04790
DGE04800

C NEGATIVE LINEAR FUNCTION OF E(K)
66 XTR3=DS CRT(TP/30.)
  SX0(1)=SX0(1) + 4*XTR3
  SX0(2)=SX0(2) + 3*XTR3
  SX0(3)=SX0(3) + 2*XTR3
  SX0(4)=SX0(4) + 1*XTR3
  GO TO 55
DGE04410
DGE04420
DGE04430
DGE04440
DGE04450
DGE04460
DGE04470
DGE04480
DGE04490
DGE04500
DGE04510
DGE04520
DGE04530
DGE04540
DGE04550
DGE04560
DGE04570
DGE04580
DGE04590
DGE04600
DGE04610
DGE04620
DGE04630
DGE04640
DGE04650
DGE04660
DGE04670
DGE04680
DGE04690
DGE04700
DGE04710
DGE04720
DGE04730
DGE04740
DGE04750
DGE04760
DGE04770
DGE04780
DGE04790
DGE04800

C
77 XTR2=DS CRT(TP/30.)
  SX0(2)=SX0(2) + XTR2
  SX0(3)=SX0(3) + 2*XTR2
  SX0(4)=SX0(4) + 3*XTR2
  SX0(5)=SX0(5) + 4*XTR2
  GO TO 55
DGE04410
DGE04420
DGE04430
DGE04440
DGE04450
DGE04460
DGE04470
DGE04480
DGE04490
DGE04500
DGE04510
DGE04520
DGE04530
DGE04540
DGE04550
DGE04560
DGE04570
DGE04580
DGE04590
DGE04600
DGE04610
DGE04620
DGE04630
DGE04640
DGE04650
DGE04660
DGE04670
DGE04680
DGE04690
DGE04700
DGE04710
DGE04720
DGE04730
DGE04740
DGE04750
DGE04760
DGE04770
DGE04780
DGE04790
DGE04800

C DC FUNCTION OF E(K)
88 XTR1=DS CRT(TP/5.)
  DO 317 I=1,M3
317 SX0(I)=SX0(I) + XTR1
DGE04410
DGE04420
DGE04430
DGE04440
DGE04450
DGE04460
DGE04470
DGE04480
DGE04490
DGE04500
DGE04510
DGE04520
DGE04530
DGE04540
DGE04550
DGE04560
DGE04570
DGE04580
DGE04590
DGE04600
DGE04610
DGE04620
DGE04630
DGE04640
DGE04650
DGE04660
DGE04670
DGE04680
DGE04690
DGE04700
DGE04710
DGE04720
DGE04730
DGE04740
DGE04750
DGE04760
DGE04770
DGE04780
DGE04790
DGE04800

95 WRITE(6,62)
62 FORMAT(IX,'SAMPLING UP TO M3 PCINT COMPLETED...')
C
MP=M3+1
DO 30 I=MP,512
  X9=C.
  X8=C.
  DO 31 J=1,NN1
    X8=SR1(J)*DCOS(2*PI*SP2(J)*DSIN(2*PI*SP2(J)*T9)
    X9=X9+X8*CEXP(SPI(J)*T9)
31
DGE04410
DGE04420
DGE04430
DGE04440
DGE04450
DGE04460
DGE04470
DGE04480
DGE04490
DGE04500
DGE04510
DGE04520
DGE04530
DGE04540
DGE04550
DGE04560
DGE04570
DGE04580
DGE04590
DGE04600
DGE04610
DGE04620
DGE04630
DGE04640
DGE04650
DGE04660
DGE04670
DGE04680
DGE04690
DGE04700
DGE04710
DGE04720
DGE04730
DGE04740
DGE04750
DGE04760
DGE04770
DGE04780
DGE04790
DGE04800

```





```

C      SX0(1)=XS
      I9=I5+I0
3C CONTINUE
C      2 WRITE(6,64)
64 FORMAT(1X,' SAMPLING UP TO 512 POINT COMPLETED....',
        *//, 'TYPE <G> TO GO ',
        1X, ' IANS
51 READ(5,51) IANS
  FORMAT(A1)
  IF(IANS.EQ.10) GO TO 1
  IF(IANS.NE.10) GO TO 2
C      RETURN
      ENC
C
C      =====
C      = SUBROUTINE TRAP - CALCULATES PEAK AND AVERAGE SIGNAL POWER =
C      =====
C      SUBROUTINE TRAP(SX0,AVG,PEAK)
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION SX0(512)
C
C      PSUM=0.
C      FIRST=SX0(1)
C      PSUM=SXC(1)**2
C      DO 10 I=2,512
C        XIM2=SX0(I)**2
C        XIM1=SX0(I-1)**2
C        PSUM=PSUM+XIM2
C        IF(XIM2.GT.XIM1) FIRST=SX0(I)
1C
C      AVG=PSUM/511.
C      PEAK=FIRST**2
C      RETURN
C      ENC
C
C      =====
C      = SUBROUTINE GAUSS - EVALUATES 5000 PCINTS GAUSSIAN DISTRIBUTION =
C      = IN ROUGH MANNER. =
C      =====
C      SUBROUTINE GAUSS(AGDF,IWA,IGA)

```











```

C
DIMENSION AGDF(5001)
IGA=IWA-1
IF(AGDF(1).EQ.SRN) GO TO 99
IF(AGDF(IWA).LE.SRN) GO TO 98
C
L=0
TEMP=IGF*.2
L2=INT(TEMP)
L1=L2
12 IF(AGDF(L1).EQ.SRN) GOTO 88
IF(AGDF(L1).GT.SRN) GOTO 22
L=L1
L1=L1+L2
GO TO 12
C
22 TEMP=L2*.2
L3=INT(TEMP)
L1=L+L3
23 IF(AGDF(L1).EQ.SRN) GO TO 88
IF(AGDF(L1).GT.SRN) GO TO 33
L=L1
L1=L1+L3
GO TO 23
C
33 TEMP=L3*.2
L4=INT(TEMP)
L1=L+L4
34 IF(AGDF(L1).EQ.SRN) GO TO 88
IF(AGDF(L1).GT.SRN) GO TO 44
L=L1
L1=L1+L4
GO TO 34
C
44 TEMP=L4*.2
L5=INT(TEMP)
L1=L+L5
45 IF(AGDF(L1).EQ.SRN) GOTO 88
IF(AGDF(L1).GT.SRN) GOTO 55
L=L1
L1=L1+L5
GO TO 45
C
55 L1=L+1
56 IF(AGDF(L1).GE.SRN) GOTO 88
L1=L1+1
GO TO 56
C

```









```

C      = SUBROUTINE NTEST - TEST AND ADJUST THE NOISE LEVEL  ROUGHFLY  =
C      =====
C      SUBROUTINE NTEST(IR,W1,W2,Q,Q1,XNO,NFLAG,N2)
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION XNO(512)
C      DATA IV,Y,I,Z/'N' /
C      234 NFLAG=0
C      IF(IR.EC.2) GO TO 2
C      POWER=XNO(1)**2
C      DO 10 I=2,512
C      10  POWER = POWER + XNO(I)**2
C      APCWR = POWER/511.
C      RATIO=C/APCWR
C      IF(RATIO.GT.10E+6) RATIO = 10E+6
C      SNRDB=10*DLOG10(RATIO)
C      1 WRITE(6,612) N2,SNRDB,W1
C      612 FORMAT(1X,'RESULTED SNR(DB) FOR ',I4,' AVERAGINGS ',F10.6,' DB
C      *//,1X,'DO YOU WANT TO READJUST THE NOISE LEVEL TO ',F10.6,' DB
C      *N<Y/N>')
C      READ(5,51) IANS
C      51  FORMAT(A1)
C      IF(IANS.EG.IZ) GO TO 19
C      IF(IANS.NE.IV) GO TO 1
C      DIF=SNRDB-W1
C      IF(DIF.GE.0.) GC TC 4
C      COEFF=-1./10**(-DIF/20.)
C      GO TO 5
C      4  COEFF= 10**(DIF/20.)
C      5  CONTINUE
C      AMAG=DSRT(APCWR)
C      BIAS=AMAG*COEFF
C      DO 20 I=1,512
C      IF(XNO(I).GE.0.0) XNO(I)=XNO(I)+BIAS
C      XNO(I)=XNO(I)-BIAS
C      20  GO TO 5
C      30  GO TO 5
C      2  PMAX=XNC(1)**2
C      DO 20 I=2,512
C      TMAX=XNO(I)**2
C      IF(TMAX.GT.PMAX) PMAX=TMAX
C      20
C
C      CGE07210
C      CGE07220
C      CGE07230
C      CGE07240
C      CGE07250
C      CGE07260
C      CGE07270
C      CGE07280
C      CGE07290
C      CGE07300
C      CGE07310
C      CGE07320
C      CGE07330
C      CGE07340
C      CGE07350
C      CGE07360
C      CGE07370
C      CGE07380
C      CGE07390
C      CGE07400
C      CGE07410
C      CGE07420
C      CGE07430
C      CGE07440
C      CGE07450
C      CGE07460
C      CGE07470
C      CGE07480
C      CGE07490
C      CGE07500
C      CGE07510
C      CGE07520
C      CGE07530
C      CGE07540
C      CGE07550
C      CGE07560
C      CGE07570
C      CGE07580
C      CGE07590
C      CGE07600
C      CGE07610
C      CGE07620
C      CGE07630
C      CGE07640
C      CGE07650
C      CGE07660
C      CGE07670
C      CGE07680

```







```

R2(3) = 5
P1(3) = -2.0
R1(4) = 5
R2(4) = -0.5
P1(4) = -2.0
P2(4) = -2.0

R1(5) = 5
R2(5) = 0.25
P1(5) = -3.0
P2(5) = -3.0
R1(6) = 5
R2(6) = 0.25
P1(6) = -3.0
P2(6) = -3.0
RETURN
END

```

C

```

08170
08180
08190
08200
08210
08220
08230
08240
08250
08260
08270
08280
08290
08300
08310
08320
08330
08340

```





# NON-LINEAR PARAMETER OPTIMIZATION PROGRAM

76



```

C READ DATA FILE
  READ(IFILE,8) NT1,NT2,NW1,NW2,NC1,NC2,NF1,NF2,N2,T2,S1
  DO 10 I=1,12,4
  10 READ(IFILE,9) XO(I),XO(I+1),XO(I+2),XO(I+3)
  8 FORMAT(2X,2A4,/,2X,2A4,/,2X,2A4,/,2X,14,/,2X,F14.8,
    */,2X,F14.8)
  5 FORMAT(2X,4(2X,F14.8))

C INFORMATION
  WRITE(6,62)
  62 FORMAT(1X,'SIGNAL DATA FILE WERE READ...',
    */,TYPE<G> TO GO.)
  READ(5,51) IANS

C PARAMETER DEFAULTED (USER DEFALBE)
  MAXFN=500
  IOPT=1
  NSIG=3
  EPS=0.0
  DELTA=0.0

C GET THE USER-DEFINED PARAMETERS
  M3 - INSTANT AT WHICH THE CCNSTANT RESIDUE FUNCTION STARTS
  N1 - NUMBER OF POLES
  NE1 - INSTANT AT WHICH THE FIRST DATA ELEMENTS IN A DATA WINDOW

  CALL PARAS(M3,N1,NE1)

C VARIABLE & DIMENSION PARAMETER
  912 CALL FRTCMS('CLRSCRN ')

C INFORMATION OF STORAGE FOR DATA WRITING
  WRITE(6,31)
  311 FORMAT(1X,'ENTER THE FILE NUMBER OF THE DATA FILE TO WHERE THE DATA
    *A ARE STORED./,
    */,1X,20-29:RESERVED FOR H/LOW DAMPING
    */,1X,20-39:
    */,1X,30-49:
    */,1X,40-59:
    */,1X,50-69:
    */,1X,60-79:
    */,1X,70-79:
    *0-WITH 0 EXTRA PARAMETERS
    *1-WITH 2 EXTRA PARAMETERS
    *2-WITH 4 EXTRA PARAMETERS
    *3-WITH 6 EXTRA PARAMETERS
    *4-WITH 8 EXTRA PARAMETERS
    NO NCISE SIGNAL PROCESSING,
    30DB SNR SIGNAL PROCESSING,
    15DB SNR SIGNAL PROCESSING,
    NO NCISE SIGNAL PROCESSING,
    30DB SNR SIGNAL PROCESSING,
    15DB SNR SIGNAL PCINTS,
    E(K):2N+2,,
    E(K):2N+4,,
    E(K):2N+6,,
    E(K):2N+8,,
    PR00C490
    PR00C500
    PR00C510
    PR00C520
    PR00C530
    PR00C540
    PR00C550
    PR00C560
    PR00C570
    PR00C580
    PR00C590
    PR00C600
    PR00C610
    PR00C620
    PR00C630
    PR00C640
    PR00C650
    PR00C660
    PR00C670
    PR00C680
    PR00C690
    PR00C700
    PR00C710
    PR00C720
    PR00C730
    PR00C740
    PR00C750
    PR00C760
    PR00C770
    PR00C780
    PR00C790
    PR00C800
    PR00C810
    PR00C820
    PR00C830
    PR00C840
    PR00C850
    PR00C860
    PR00C870
    PR00C880
    PR00C890
    PR00C900
    PR00C910
    PR00C920
    PR00C930
    PR00C940
    PR00C950
    PR00C960
  
```



[illegible]















```

C      IMPLICIT REAL*8(A-H,O-Z)
C      DATA IG/,G/,IB/,B/,
C      1 CALL FRTCMS('CLRSCRN ')
      WRITE(6,61)
      READ(5,*) N1
      WRITE(6,62)
      READ(5,*) N1
      WRITE(6,63)
      READ(5,*) M3
61  FORMAT(1X,'ENTER THE NUMBER OF POLES OF THE RESPONSE (# OF POLES)')
62  FORMAT(1X,'HOW MANY EARLY TIME DATA POINTS DO YOU WANT TO HAVE?')
63  */
      1X,'IN YOUR CURRENT DATA WINDOW? : 0 UP TO 10 (MAX)')
      1X,'SPECIFY K* POINT AT WHICH E(K) IS ASSUMED TO BE ZERO'
      1X,'MATCH IT TO YOU K* OF YOUR DATA GENERATION ROUTINE'
C      CHANGE IF NECESSARY
2  WRITE(6,64)
64  FORMAT(1X,'TYPE <B> TO GO BACK TO CHANGE THE ABOVE VARIABLES',
      */
      1X,'OTHERWISE TYPE <G> TO GO')
51  READ(5,*) IANS
      FORMAT(A1)
      IF(IANS.EQ.18) GO TO 1
      IF(IANS.NE.18) GO TO 2
C      RETURN
C      END
=====
= SUBROUTINE INGU : INPUT THE INITIAL TRIALS FOR THE UNKNOWNS =
=====
SUBROUTINE INGU(X,N,N1,N1)
  IMPLICIT REAL*8(A-H,O-Z)
  DATA IV/,V/,IZ/,N/,
  DIMENSION X(N)
C      1 CALL FRTCMS('CLRSCRN ')
C      61  WRITE(6,61)
      10=0
      DO 10 I=1,N1,4

```

















```

PR004330
PR004340
PR004350
PR004360
PR004370
PR004380
PR004390
PR004400
PR004410
PR004420
PR004430
PR004440
PR004450
PR004460
PR004470
PR004480
PR004490
PR004500
PR004510
PR004520
PR004530
PR004540
PR004550
PR004560
PR004570
PR004580
PR004590
PR004600
PR004610
PR004620
PR004630
PR004640
PR004650
PR004660
PR004670
PR004680
PR004690
PR004700
PR004710
PR004720
PR004730
PR004740
PR004750
PR004760
PR004770
PR004780
PR004790
PR004800

```

# INITIALIZE VARIABLES

```

C
ISCALL = ISCALL+1
ISCALLU = ISCALL+N
IXNEW1 = ISCALLU
IXNEW1+1 = IXNEW1+1
IXBADI = IXNEW1+N
IFPL1 = IFPL1+1
IFPL1+1 = IFPL1+M
IFPL1 = IFPL1+M
IFML1 = IFML1+1
IFML1+1 = IFML1+1
IMJC = IXJAC - M

AL = CNE
CONS2 = TENTH
IF (IOPT.EQ.0) GO TO 20
IF (IOPT.EQ.1) GO TO 10
AL = PARM(1)
FO = PARM(2)
UP = PARM(3)
CONS2 = PARM(4)*HALF
GO TO 15

1C AL = PC1
FO = TWC
UP = HUNT

15 ONESFO = ONE/FO
FOSQ = FO*FO
FOSQS4 = FOSQ**4
20 IEVAL = 0
DELTA2 = DELTA*HALF
ERL2 = CNEP10
IBAD = -99
ISH = 1
ITER = -1
INFER = 0
IER = C
DO 25 J=IDELXL, IDELXU
  WORK(J) = ZERO
25 CONTINUE
GO TO 165

C
3C SSQOLD = SSQ
C
IF (INFER.GT.0.OR. IJAC.GE.N.OR. ICCUNT.GT.0) GO TO 55
RANK ONE UPDATE TO JACOBIAN
IJAC = IJAC+1
DSQ = ZERO
DO 35 J=IDELXL, IDELXU
  DSQ = DSC+WORK(J)*WCRK(J)
35

```



```

35 CONTINUE
   IF (DSC.LE.ZERO) GC TO 55
   DO 50 I=1,M
      G = F(I)-WORK(IFML1+1)
      K = I
      DO 40 J=IDELXL, IDELXU
         G = G+XJAC(K)*WORK(J)
         K = K+IXJAC
      CONTINUE
      G = G/DSC
      K = I
      DO 45 J=IDELXL, IDELXU
         XJAC(K) = XJAC(K)-G*WORK(J)
         K = K+IXJAC
      CONTINUE
      GO TO 80
50 CONTINUE
      GO TO 80
C   55 IJAC = C
      K = -IJAC
      DO 75 J=1,N
         K = K+IJAC
         XDABS = DABS(X(J))
         HH = REL*(DMAX1(XDABS,AXI))
         XHOLD = X(J)
         X(J) = X(J)+HH
      CONTINUE
      CALL FUNC(X,M,N,WORK(IFPL1,T2,M3,N1,NE1))
      IEVAL = IEVAL+1
      X(J) = XFOLD
      IF (ISW.EQ.1) GC TO 65
C
C   X(J) = XHOLD-HH
      CONTINUE
      CALL FUNC(X,M,N,WORK(IFML1,T2,M3,N1,NE1))
      IEVAL = IEVAL+1
      X(J) = XFOLD
      RHH = HALF/HH
      DO 60 I=IFPL, IFPU
         K = K+1
         XJAC(K) = (WORK(I)-WORK(I+M))*RHH
      CONTINUE
      GO TO 75
60 CONTINUE
C   65 RHH = ONE/HH
      DO 70 I=1,M

```

JACOBIAN BY INCREMENTING X

CENTRAL DIFFERENCES

FORWARD DIFFERENCES

FR004610  
 PR004620  
 FR004630  
 PR004640  
 FR004650  
 PR004660  
 FR004670  
 PR004680  
 FR004690  
 PR004700  
 FR004710  
 PR004720  
 FR004730  
 PR004740  
 FR004750  
 PR004760  
 FR004770  
 PR004780  
 FR004790  
 PR004800  
 FR004810  
 PR004820  
 FR004830  
 PR004840  
 FR004850  
 PR004860  
 FR004870  
 PR004880  
 FR004890  
 PR004900  
 FR004910  
 PR004920  
 FR004930  
 PR004940  
 FR004950  
 PR004960  
 FR004970  
 PR004980  
 FR004990  
 PR005000  
 FR005010  
 PR005020  
 FR005030  
 PR005040  
 FR005050  
 PR005060  
 FR005070  
 PR005080  
 FR005090  
 PR005100  
 FR005110  
 PR005120  
 FR005130  
 PR005140  
 FR005150  
 PR005160  
 FR005170  
 PR005180  
 FR005190  
 PR005200  
 FR005210  
 PR005220  
 FR005230  
 PR005240  
 FR005250  
 PR005260  
 FR005270  
 PR005280



```

C
7C      K = K+1
75      XJAC(K) = (WCRK((IFPL1+I)-F(I))*RHF
          CONTINUE
          CALCULATE GRADIENT
80      ERL2X = ERL2
          ERL2 = ZERO
          K = -IMJC
          DO 90 J=IGRADL, IGRADU
              K = K+IMJC
              SUM = ZERO
              DO 85 I=1,M
                  K = K+1
                  SUM = SUM+XJAC(K)*F(I)
              CONTINUE
              WORK(J) = SUM
              ERL2 = ERL2+SUM*SUM
          CONTINUE
          ERL2 = CSQRT(ERL2)
          IF (I JAC.GT.0) GO TO 95
          IF (ERL2.LE.DELTA2) INFER = INFER+4
          IF (ERL2.LE.CONVS2) ISW = 2
          CONVERGENCE TEST FOR NURM OF GRADIENT
          CALCULATE THE LOWER SUPER TRIANGE OF
          JACOBIAN (TRANSPPOSED) * JACOBIAN
          C
95      L = 0
          IS = -IXJAC
          DO 110 I=1,N
              IS = IS+IXJAC
              JS = -IXJAC
              DO 105 J=1,I
                  JS = JS+IXJAC
                  L = L+1
              SUM = ZERC
              CC 100 K=1,M
                  LI = JS+K
                  LJ = JS+K
                  SUM = SUM+XJAC(LI)*XJAC(LJ)
              CONTINUE
              XJTJ(L) = SUM
          CONTINUE
          CONVERGENCE CHECKS
          C
100      IF (INFER.GT.0) GO TO 315
105      IF (IEVAL.GE.MAXFN) GO TO 290
110      IF (IGFT.EQ.0) GO TO 120
          K = 0
          COMPUTE SCALING VECTOR
          C

```





```

DO 115 J=1,N
  K = K+J
  WORK(ISCAL1+J) = XJTJ(K)
115 CONTINUE
GO TO 125

C 120 DNORM = ZERG
  K = 0
DO 125 J=1,N
  K = K+J
  WORK(ISCAL1+J) = DSQRT(XJTJ(K))
  DNORM = DNORM+XJTJ(K)*XJTJ(K)
125 CONTINUE
DNORM = ONE/DSQRT(DNORM)

C      NORMALIZE SCALING VECTOR

DO 130 J=ISCAL1,ISCALU
  WORK(J) = WORK(J)*DNORM*ERL2
130 CONTINUE

C      ADD L-M FACTOR TO DIAGONAL

135 ICOLNT = 0
140 K = 0
DO 150 I=1,N
  DO 145 J=1,I
    K = K+1
    WCRK(K) = XJTJ(K)
145 CONTINUE
    WORK(K) = WORK(K)+WORK(ISCAL1+I)*AL
    WORK(ICELX1+I) = WORK(ICELX1+I)
150 CONTINUE

C      CHOLESKY DECOMPOSITION

155 CALL LECTIP (WORK,1,N,WORK(IDELX1),N,0,G,XHCLD,IER)
IF (IER.EQ.0) GO TO 160
IER = G
IF (I JAC.GT.0) GO TC 55
IF (IBAL.LE.0) GO TO 240
IF (IBAL.GE.2) GO TC 310
GO TO 150
160 IF (IBAL.NE.-99) IBAD = 0

C      CALCULATE SUM CF SQUARES

165 DO 170 J=1,N
  WORK(IXNEW1+J) = X(J)-WORK(IDELX1+J)
170 CONTINUE

C  CALL FUNC
  CALL FUNC (WORK(IXNEW1),M,N,WORK(IFPL),T2,M3,N1,NE1)
  IEVAL = IEVAL+1
  SSC = ZERG
DO 175 I=IFPL,IFPU

```

```

PR005770
PR005780
FR005790
PR005800
FR005810
PR005820
FR005830
PR005840
FR005850
PR005860
FR005870
PR005880
FR005890
FR005900
PR005910
FR005920
PR005930
FR005940
PR005950
FR005960
FR005970
PR005980
FR005990
FR006000
PR006010
FR006020
PR006030
FR006040
FR006050
FR006060
PR006070
FR006080
FR006090
PR006100
FR006110
PR006120
FR006130
FR006140
FR006150
FR006160
PR006170
FR006180
FR006190
FR006200
FR006210
FR006220
PR006230
FR006240

```



```

      SSQ = SSC+WORK(I)*WORK(I)
175 CONTINUE
      IF (ITER.GE.0) GO TO 185
C
      ITER = C
      SSQOLD = SSQ
      DO 180 I=1,M
        F(I) = WCRK(IFPL1+I)
180 CONTINUE
      GO TO 185
185 IF (IOPT.EQ.0) GO TO 215
C
      IF (SSQ.LE.SSQOLD) GO TO 205
C
190 ICOUNT = ICOUNT+1
      AL = AL*FOSC
      IF (IJAC.EQ.0) GC TO 195
      IF (ICCLNT.GE.4.OR.AL.GT.UP) GC TO 200
195 IF (AL.LE.UP) GC TO 140
      IF (IBAC.EQ.1) GC TO 310
      IER = 35
      GO TO 315
200 AL = AL/FOSCS4
      GO TO 185
C
205 IF (ICCLNT.EQ.0) AL = AL/F0
      IF (ERL2X.LE.ZERO) GO TO 210
      G = (ERL2X/ERL2X)
      IF (ERL2.LT.ERL2X) AL = AL*DMAX1(ONESF0,G)
      IF (ERL2.GT.ERL2X) AL = AL*DMIN1(F0,G)
210 AL = DMAX1(AL,PREC)
C
215 ITER = ITER+1
      DO 220 J=1,N
        X(J) = WORK(IXNEW1+J)
220 CONTINUE
      DO 225 I=1,M
        WORK(IFML1+I) = F(I)
        F(I) = WCRK(IFPL1+I)
225 CONTINUE
C
      IF (AL.GT.5.0D0) GO TO 30
      DO 230 J=1,N
        XDIF = DABS(WORK(IDELX1+J))/DMAX1(DABS(X(J)),AX)
        IF (XCIF.GT.RELCON) GO TO 235
230 CONTINUE
      INFER = 1
C
      SSQ FOR INITIAL ESTIMATES OF X
      CHECK DESCENT PROPERTY
      INCREASE PARAMETER AND TRY AGAIN
      ADJUST MARQUARDT PARAMETER
      ONE ITERATION CYCLE COMPLETED
      RELATIVE CONVERGENCE TEST FOR X
      RELATIVE CONVERGENCE TEST FOR SSQ

```

```

PR006250
PR006260
PR006270
PR006280
PR006290
PR006300
PR006310
PR006320
PR006330
PR006340
PR006350
PR006360
PR006370
PR006380
PR006390
PR006400
PR006410
PR006420
PR006430
PR006440
PR006450
PR006460
PR006470
PR006480
PR006490
PR006500
PR006510
PR006520
PR006530
PR006540
PR006550
PR006560
PR006570
PR006580
PR006590
PR006600
PR006610
PR006620
PR006630
PR006640
PR006650
PR006660
PR006670
PR006680
PR006690
PR006700
PR006710
PR006720

```







```

315 G = SIG
DO 320 J=1,N
  XHOLD = DABS(WORK(IDELX1+J))
  IF (XHOLD.LE.ZERC) GO TO 320
  G = CMIN1(G,-DLOG10(XHOLD)+DLOG10(DMAX1(AX,DABS(X(J))))))
320 CONTINUE
  IF (N.GT.2) GO TO 330
  DO 325 J=1,N
    WORK(J+5) = WORK(J+IGRAD1)
325 WORK(1) = ERL2+ERL2
330 WORK(2) = IEVAL
    SSC = SSCGLC
    WORK(3) = G
    WORK(4) = AL
    WORK(5) = ITER
335 CALL UERSET(LEVOLD,LEVCLD)
  IF (IER.EQ.0) GO TO 9005
900C CONTINUE
  CALL UERTST (IER,6HZSSQ)
9005 RETURN
  END
=====
SUBROUTINE LEQTP
  PURPOSE
    - LINEAR EQUATION SOLUTION - POSITIVE DEFINITE
      MATRIX - SYMMETRIC STORAGE MODE - SPACE
      ECONOMIZER SOLUTION
=====
SUBROUTINE LEQTP (A,M,N,B,IB,IDGT,D1,D2,IER)
  DIMENSION A(1),B(1B,1)
  DOUBLE PRECISION A,B,D1,D2
  FIRST EXECUTABLE STATEMENT
  INITIALIZE IER
  IER = 0
  DECOMPOSE A
  CALL LUCECP (A,A,N,D1,D2,IER)
  IF (IER.NE.0) GO TO 9000
  DO 5 I=1,M
    CALL LUELMP (A,B(1,I),N,B(1,I))
5 CONTINUE
  GO TO 5005
900C CONTINUE
  CALL UERTST(IER,6HLEQTP)
9005 RETURN
  END
=====

```





```

315 G = SIG
DO 320 J=1,N
  XHOLD = DABS(WORK(IDELX1+J))
  IF (XHOLD.LE.ZERC) GO TO 320
  G = DMIN(G,-DLOG10(XHOLD)+DLOG10(DMAX1(AX,DABS(X(J))))))
320 CONTINUE
  IF (N.GT.2) GO TO 330
DO 325 J=1,N
  WORK(J+5) = WORK(J+IGRAD1)
325 WORK(1) = ERL2+EKL2
330 WORK(2) = IEVAL
  SSC = SSCCLD
  WORK(3) = G
  WORK(4) = AL
  WORK(5) = ITER
335 CALL UEFSET(LEVOLD,LEVCLD)
  IF (IER.EQ.0) GO TO 9005
9000 CONTINUE
  CALL UERTST (IER,6H2XSSQ)
9005 RETURN
      END
=====
SUBROUTINE LEQTP  - LINEAR EQUATION SOLUTION - POSITIVE DEFINITE
PURPOSE          MATRIX - SYMMETRIC STORAGE MODE - SPACE
                  ECONOMIZER SOLUTION
=====
SUBROUTINE LEQTP (A,M,N,B,IB,IDGT,D1,D2,IER)
  DIMENSION      A(1),B(1B,1)
  DOUBLE PRECISION A,B,D1,D2
  FIRST EXECUTABLE STATEMENT
  INITIALIZE IER
  DECOMPOSE A
  IER = 0
  CALL LUCECP (A,A,N,D1,D2,IER)
  IF (IER.NE.0) GO TO 9000
  DO 5 I=1,M
    CALL LU2ELMP (A,B(1,1),N,B(1,1))
5 CONTINUE
  GO TO 5C05
9000 CONTINUE
  CALL UERTST(IER,6HLEQTP)
9005 RETURN
      END

```











```

      T = T - A(IS) * X(KK)
      KK = KK - 1
      IS = IS - KK
      CONTINUE
25  X(II) = T * A(IS)
26  CONTINUE
27  RETURN
28  END
=====
SUBROUTINE UERSET
PURPOSE
=====
MSG :
=====
      - SET MESSAGE LEVEL FOR INSL ROUTINE UERTST
      LEVEL = 4 CAUSES ALL MESSAGES TO BE
      PRINTED, ARE PRINTED IF IER IS
      GREATER THAN 32,
      LEVEL = 3 MESSAGES ARE PRINTED IF IER IS
      GREATER THAN 64,
      LEVEL = 2 MESSAGES ARE PRINTED IF IER IS
      GREATER THAN 128,
      LEVEL = 1 MESSAGES ARE PRINTED IF IER IS
      GREATER THAN 128,
      LEVEL = 0 ALL MESSAGE PRINTING IS
      SUPPRESSED.
=====
SUBROUTINE UERSET (LEVEL,LEVELD)
INTEGER
LEVEL,LEVELD
LEVELD = LEVEL
CALL UERTST (LEVGLD,6HURSET)
RETURN
END
=====
SUBROUTINE UERTST
PURPOSE
=====
      - PRINT A MESSAGE REFLECTING AN ERROR CONDITION
      - ERROR PARAMETER. (INPUT)
      IER = I+J WHERE
      I = 128 IMPLIES TERMINAL ERROR MESSAGE,
      I = 64 IMPLIES WARNING WITH FIX MESSAGE,
      I = 32 IMPLIES WARNING MESSAGE,
      J = ERROR CODE RELEVANT TO CALLING
      ROUTINE.
      NAME - A CHARACTER STRING OF LENGTH SIX PROVIDING
      THE NAME OF THE CALLING ROUTINE. (INPUT)
=====
SUBROUTINE UERTST (IER,NAME)
=====

```

CCCCCCCCCCCCCCCC C C CCCCCCCCCCCCCCCCC





```

C          INTEGER
C          INTEGER
C          *
C          DATA
C          DATA
C          DATA
C          CALL USPDK (NAME,6,NAMUPK,MTB)
C          CALL UGETIO(1,NIN,ICUNIT)
C          IF (IER.GT.999) GO TO 25
C          IF (IER.LT.-32) GO TO 55
C          IF (IER.LE.128) GO TO 5
C          IF (LEVEL.LT.1) GO TO 30
C          IF (IECCF.EQ.1) WRITE(ICUNIT,35) IER,NAMEQ,IEQ,NAMUPK
C          IF (IECCF.EQ.0) WRITE(ICUNIT,35) IER,NAMUPK
C          GO TO 3C
C          5 IF (IER.LE.64) GO TO 10
C          IF (LEVEL.LT.2) GO TO 30
C          IF (IECCF.EQ.1) WRITE(ICUNIT,40) IER,NAMEQ,IEQ,NAMUPK
C          IF (IECCF.EQ.0) WRITE(ICUNIT,40) IER,NAMUPK
C          GO TO 3C
C          1C IF (IER.LE.32) GO TO 15
C          IF (LEVEL.LT.3) GO TO 30
C          IF (IECCF.EQ.1) WRITE(ICUNIT,45) IER,NAMEQ,IEQ,NAMUPK
C          IF (IECCF.EQ.0) WRITE(ICUNIT,45) IER,NAMUPK
C          GO TO 3C
C          15 CONTINUE
C          DO 20 I=1,6
C          IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25
C          20 CONTINUE
C          LEVCLD = LEVEL
C          LEVEL = IER
C          IER = LEVCLD
C          IF (LEVEL.LT.0) LEVEL = 4
C          IF (LEVEL.GT.4) LEVEL = 4
C          GO TO 3C
C          25 CONTINUE
C          IF (LEVEL.LT.4) GO TO 30
SPECIFICATIONS FOR ARGUMENTS
NAME(1)
I, IEQ, IEQDF, ICUNIT, LEVEL, LEVOLD, NAMEQ(6),
NAMSET(6), NAMUPK(6), NIN, MTB
NAMEQ/1HU, 1HE, 1HR, 1HS, 1HE, 1HT/
NAMEQ/6*1H /
LEVEL/4/, IEQDF/0/, IEQ/1H=/
UNPACK NAME INTO NAMUPK
FIRST EXECUTABLE STATEMENT
GET OUTPUT UNIT NUMBER
CHECK IER
PRINT TERMINAL MESSAGE
PRINT WARNING WITH FIX MESSAGE
PRINT WARNING MESSAGE
CHECK FOR UERSET CALL

```

```

PR005130
PR005140
PR005150
PR005160
PR005170
PR005180
PR005190
PR005200
PR005210
PR005220
PR005230
PR005240
PR005250
PR005260
PR005270
PR005280
PR005290
PR005300
PR005310
PR005320
PR005330
PR005340
PR005350
PR005360
PR005370
PR005380
PR005390
PR005400
PR005410
PR005420
PR005430
PR005440
PR005450
PR005460
PR005470
PR005480
PR005490
PR005500
PR005510
PR005520
PR005530
PR005540
PR005550
PR005560
PR005570
PR005580
PR005590
PR005600

```



```

C          PRINT NON-DEFINED MESSAGE
          IF (IECCF.EQ.1) WRITE(IUNIT,50) IER,NAMEQ,IEQ,NAMUPK
          IF (IECCF.EQ.0) WRITE(IUNIT,50) IER,NAMUPK
          IEQCF = 0
          RETURN
          30 FORMAT (15H *** TERMINAL ERROR,10X,7H( IER = ,I3,
1          20H) FROM IMSL ROUTINE ,6A1,6A1)
          40 FORMAT (27H *** WARNING WITH FIX ERROR,2X,7H( IER = ,I3,
1          20H) FROM IMSL ROUTINE ,6A1,6A1)
          45 FORMAT (18H *** WARNING ERROR,11X,7H( IER = ,I3,
1          20H) FROM IMSL ROUTINE ,6A1,6A1)
          50 FORMAT (20H *** UNDEFINED ERROR,9X,7H( IER = ,I5,
1          20H) FROM IMSL ROUTINE ,6A1,6A1,6A1)

          SAVE P FOR P = R CASE
          P IS THE PAGE NAMUPK
          R IS THE ROUTINE NAMUPK

          55 IEQDF = 1
          DO 60 I=1,6
          60 NAMEQ(I) = NAMUPK(I)
          65 RETURN
          65 ENC

=====
SUBROUTINE UGETIO
PURPOSE
=====
- TO RETRIEVE CURRENT VALUES AND TO SET NEW
  VALUES FOR INPUT AND OUTPUT UNIT IDENTIFIERS.
=====
ARGUMENTS      IOPT
=====
1. IF IOPT=1, THE CURRENT INPUT AND OUTPUT
   UNIT IDENTIFIERS ARE RETURNED IN NIN
   AND NOUT, RESPECTIVELY.
2. IF IOPT=2, THE INTERNAL VALUE OF NIN IS
   RESET FOR SUBSEQUENT USE.
3. IF IOPT=3, THE INTERNAL VALUE OF NOUT IS
   RESET FOR SUBSEQUENT USE.
- INPUT UNIT IDENTIFIER.
  NIN
- OUTPUT UNIT IDENTIFIER.
  NOUT
- OUTPUT IF IOPT=1, INPUT IF IOPT=2.
- OUTPUT IF IOPT=1, INPUT IF IOPT=3.

SUBROUTINE UGETIO(ICPT,NIN,NOUT)
SPECIFICATIONS FOR ARGUMENTS
INTEGER      IOPT,NIN,NOUT
SPECIFICATIONS FOR LOCAL VARIABLES
INTEGER      NIND,NOUTD
DATA         NIND/57,NOUTD/6/
FIRST EXECUTABLE STATEMENT
=====

```







```

110 J = J+1
C
C      CHECK UNPAKD ARRAY AND SET NCHMTB
C      BASED ON TRAILING BLANKS FOUND
      DO 200 N = 1, NWORDS, 4
        NN = NWORDS - N - 2
        LBYTE = UNPAKD(NN)
        IF (LBYTE .NE. IBLANK) GO TO 210
      200 CONTINUE
      210 NCHMTB = (NN + 3) / 4
      RETURN
      END
C
C      =====
C      = SUBROUTINE DATA - SUPPLIES THE INITIAL GUESSES WITH A FILE
C      =====
C
C      SUBROUTINE DATA(X,N,N1,NE1)
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION X(N)
C
C      PREDEFINED INITIAL GUESS
      X(1) = 1.1
      X(2) = 0.9
      X(3) = -1.1
      X(4) = -0.9
      X(5) = 1.1
      X(6) = -0.9
      X(7) = -1.1
      X(8) = -0.9
      X(9) = -0.55
      X(10) = 0.55
      X(11) = -1.8
      X(12) = 2.2
      X(13) = 0.45
      X(14) = -0.55
      X(15) = -1.8
      X(16) = -2.2
C
      X(17) = 0.225
      X(18) = 0.275
      X(19) = -2.8
      X(20) = 3.2
      X(21) = 0.225
      X(22) = -0.275
      X(23) = -3.8
      X(24) = -3.2
C
PR01 C570
PR01 C580
PR01 C590
PR01 C600
PR01 C610
PR01 C620
PR01 C630
PR01 C640
PR01 C650
PR01 C660
PR01 C670
PR01 C680
PR01 C690
PR01 C700
PR01 C710
PR01 C720
PR01 C730
PR01 C740
PR01 C750
PR01 C760
PR01 C770
PR01 C780
PR01 C790
PR01 C800
PR01 C810
PR01 C820
PR01 C830
PR01 C840
PR01 C850
PR01 C860
PR01 C870
PR01 C880
PR01 C890
PR01 C900
PR01 C910
PR01 C920
PR01 C930
PR01 C940
PR01 C950
PR01 C960
PR01 C970
PR01 C980
PR01 C990
PR01 C1000
PR01 C1010
PR01 C1020
PR01 C1030
PR01 C1040

```





PROJ1C50  
PROJ1C60

RETURN  
END



APPENDIX C  
DATA SIGNAL PLOTS

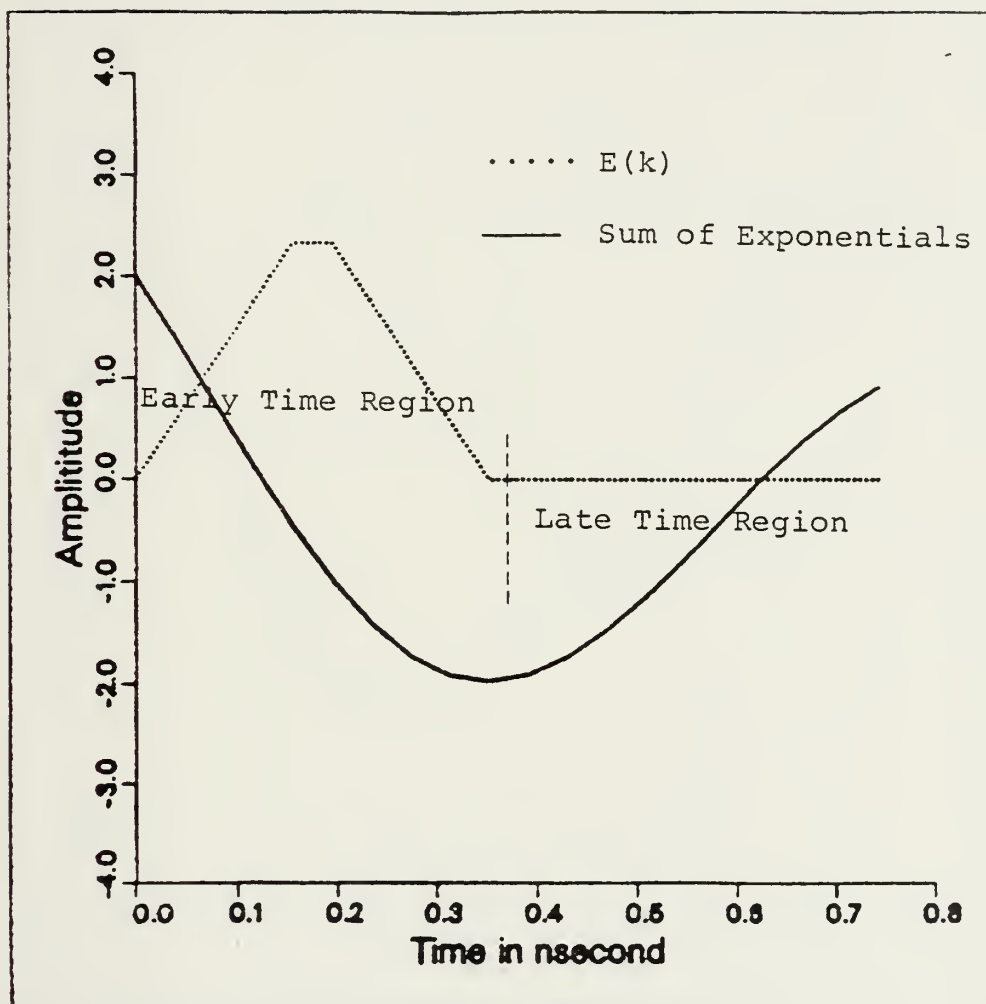


Figure C.1. Decomposition of Signal 1  
(Noise Free)



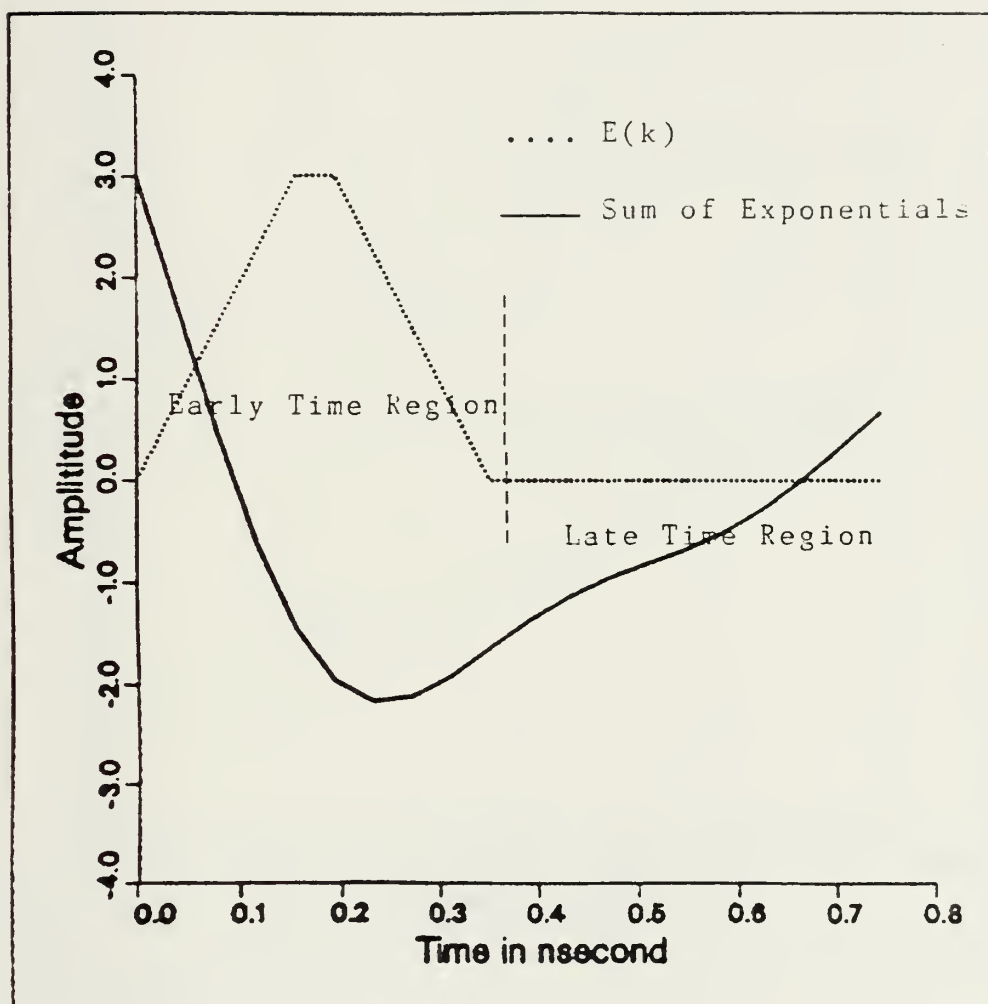


Figure C.2. Decomposition of Signal 1  
(Noise Free)



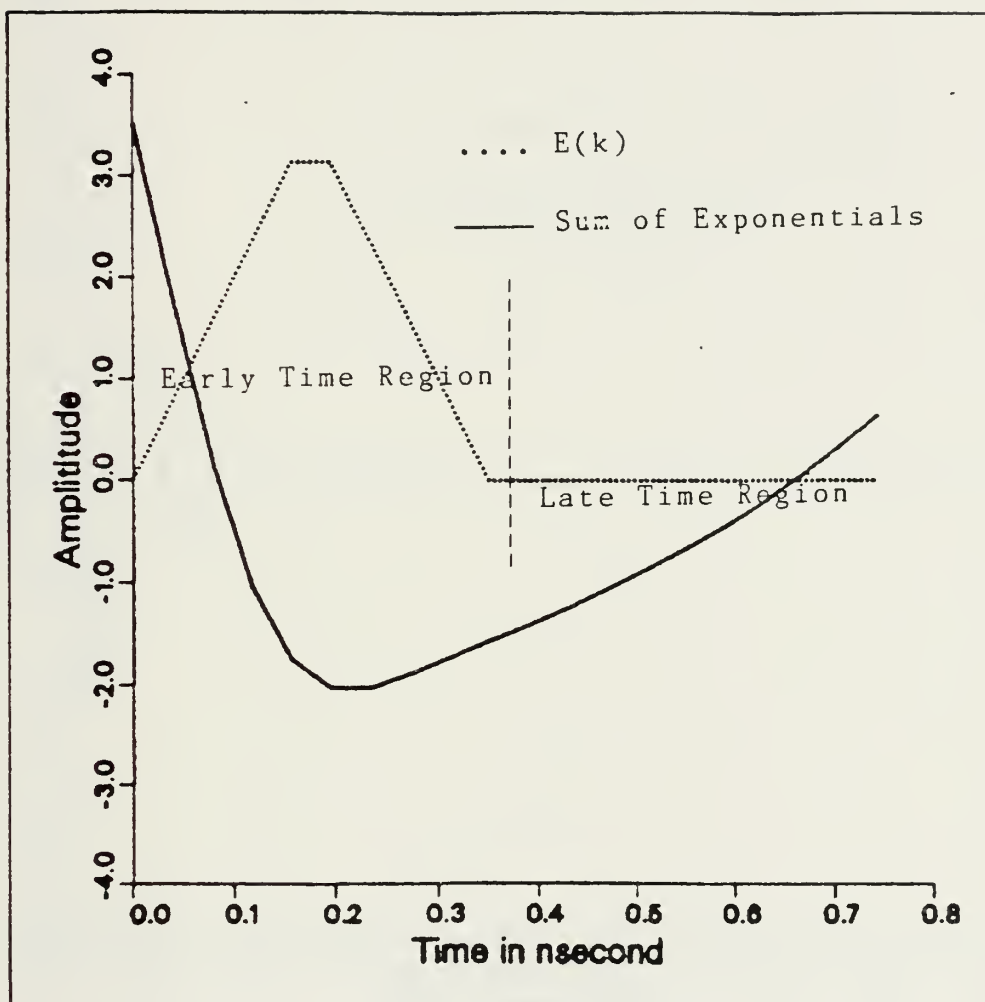


Figure C.3. Decomposition of Signal 3  
(Noise Free)





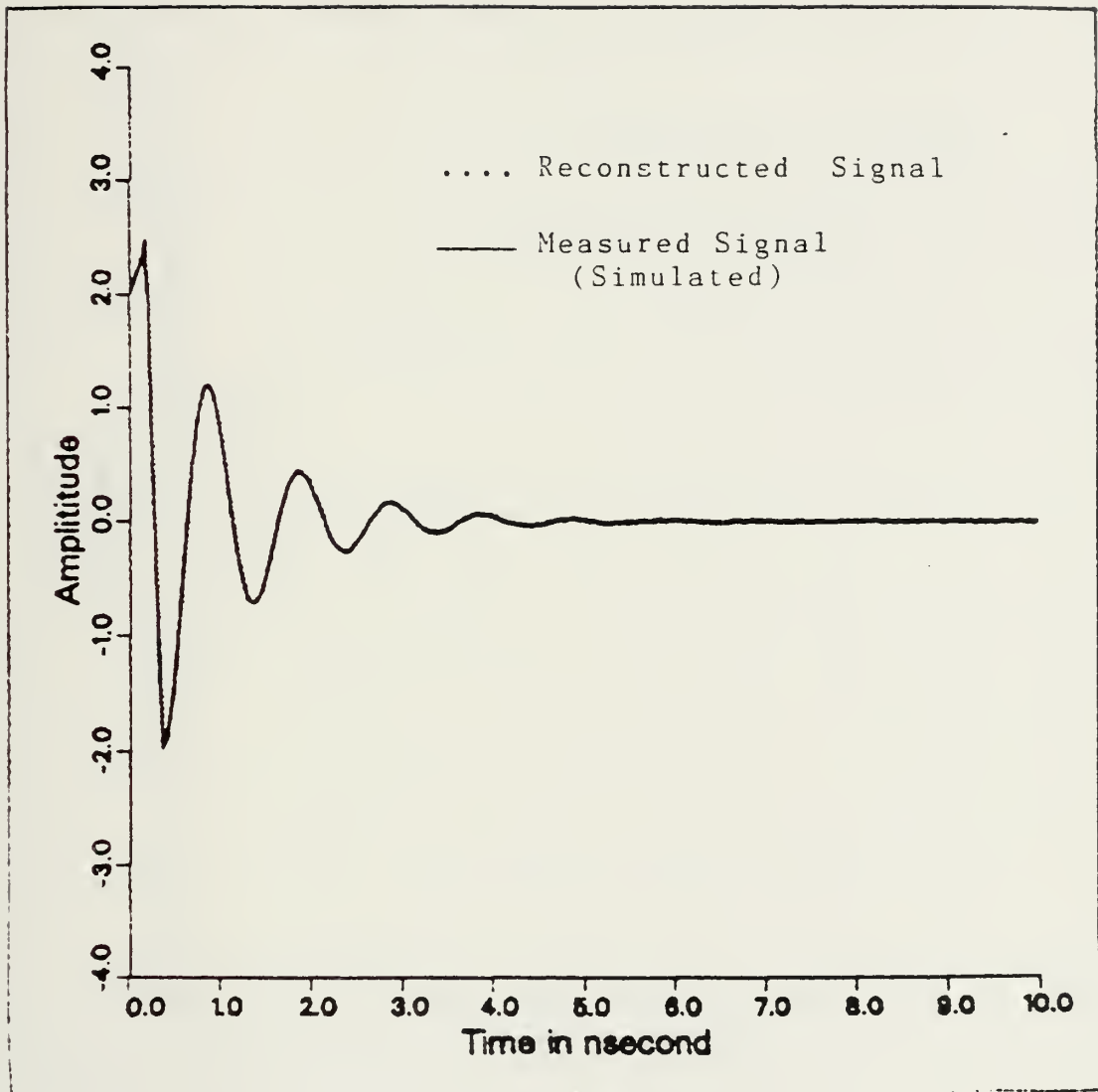


Figure C.4. Reconstruction of Signal 1 from The  
Computed Poles and Residues(SNR=30db)



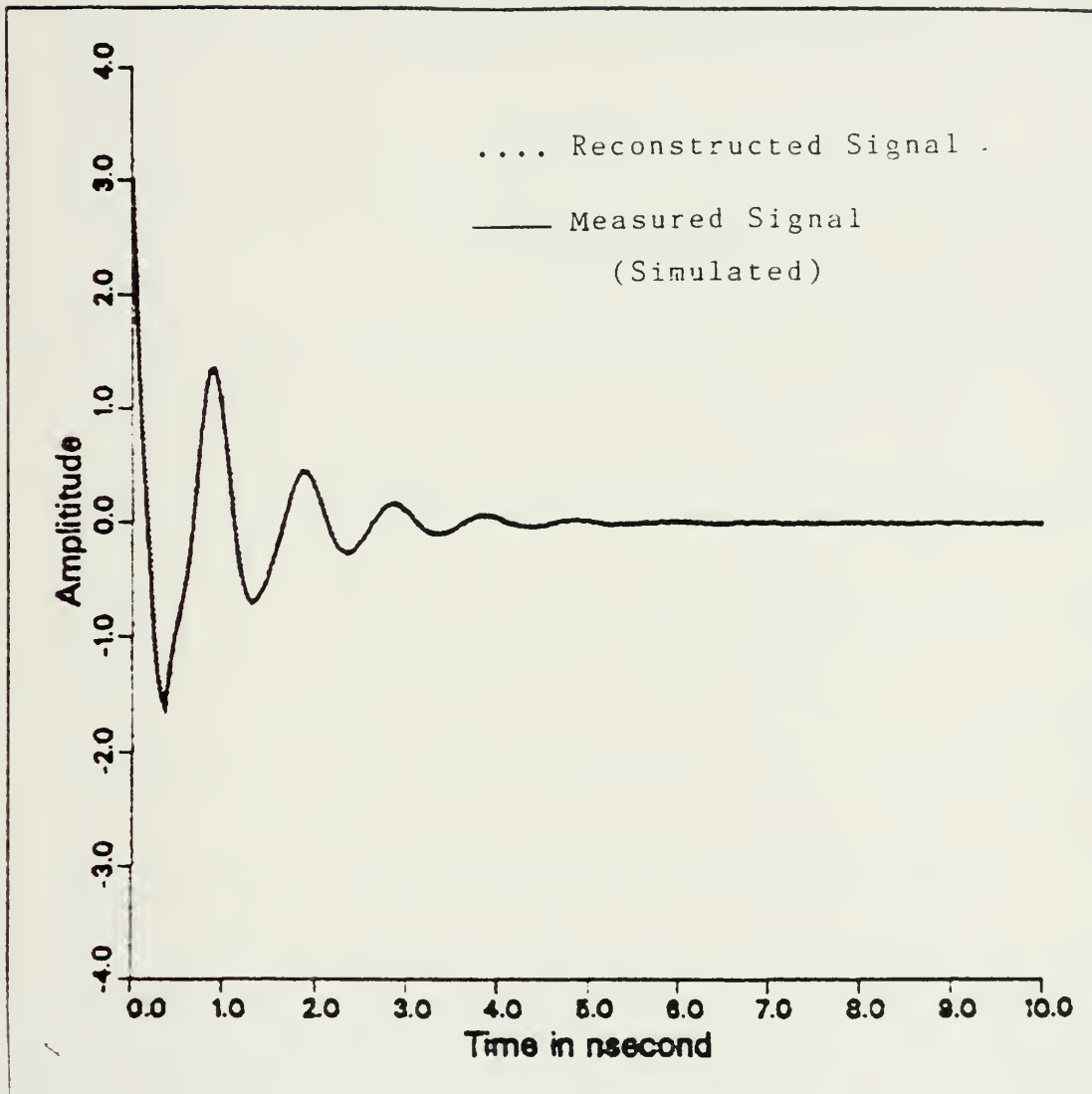


Figure C.5. Reconstruction of Signal 2 from The  
Computed Poles and Residues(SNR=30db)



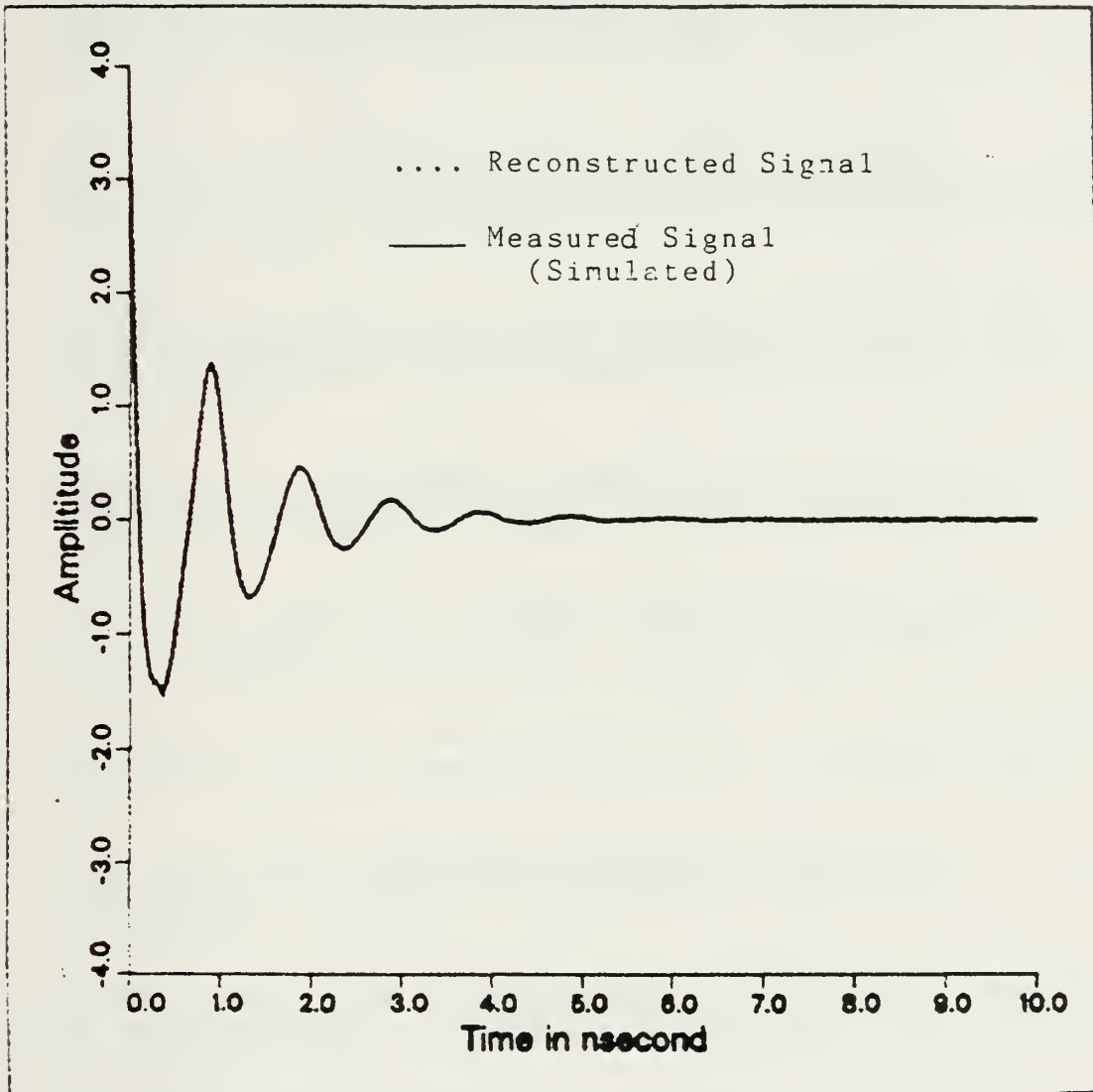


Figure C.6. Reconstruction of Signal 3 from The  
Computed Poles and Residues(SNR=30db)



## LIST OF REFERENCES

1. Baum, C.E., Singularity Expansion of Electromagnetic Fields and Potentials Radiated From Antennas of Scattered From Objects in Free Space, Sensor and Simulation Notes, Note 179, Air Force Weapon Laboratory, May 1973.
2. Mains, R.H., and Moffatt, D.L., Complex Natural Resonances of an Object in Detection and Discrimination, Ohio State University, Electro Science Laboratory, Columbus, Ohio, June 1974.
3. Manilha, Fernando M.C., Investigation of Method for Natural Response Target Identification, Master Thesis, Naval Postgraduate School, Monterey, California, December 1982.
4. Morgan, M.A., Singularity Expansion Representation of Fields and Currents in Transient Scattering, Naval Postgraduate School, Monterey, California, 1983.
5. Davenport, Richard W., Target Identification Using Prony's Method, Master Thesis, Naval Postgraduate School, Monterey, California, October 1982.
6. Marquardt, D., H., "An Algorithm For the Least-Squares Estimation of Non-Linear Parameters", J. SIAM, 11(2), 1963.
7. Bard, Yonathan, Non-Linear Parameter Estimation, International Business Corporation, Cambridge, Massachusetts, 1974.
8. Bennett, Leonard C., "Time-Domain Electromagnetics and Its Applications", Proceeding of the IEEE, Vol. 66, No. 3, March 1978.
9. Levenberg, K., "A Method for the Solution of Certain Non-Linear Problems in Least Squares", QUART. APPL. MATH 2, 1944.
10. Van Blaricum, M.L., and Mittra, R., Techniques for Extracting the Complex Resonances of a System Directly From Its Transient Response, AFWL-TR-76-250, University of Illinois, Electromagnetics Laboratory, Department of Electrical Engineering, Urbana, Illinois, April 1977.





11. Prony, R., Essai Experimental et Analytique sur les Lois de la Dilatibilitie des Fluides Elastiques et sur Celles de la Force Expansive de la Vapeur de l'AIK00I a Differentes Temperatures, J. l'Ecole Polytech. (Paris), Vol. 1, No. 2, pp. 24-76, 1975.
12. Householder, A.S., On Prony's Method of Fitting Exponential Decay Curves and Multiple-Hit Survival Curves, Oak Ridge National Laboratory, Report ORNL-455, Oak Ridge, Tennessee, February 1950.
13. The IMSL Library Volume 4 - International Mathematical & Statistical Libraries Inc., Houston, Texas, June 1982.
14. DISSPLA - Display Integrated Software System and Plotting Language Pocket Guide, Integrated Software Systems Corporation, San Diego, California, 1981.



# INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3. Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
4. Professor M. A. Morgan, Code 62Mw Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	10
5. Professor Hung-Mou Lee, Code 62Lm Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
6. LCDR Choong Yeon Chong 100-33, Kyo-Dong, Chun-Cheon SI, Kangwondo, Seoul, Korea	3
7. Division Chief Education Division Department of Personnel and Administration R.O.K. Navy Headquarters Seoul, Korea	1







205817

Thesis  
C448872 Chong  
c.1

Investigation of non-  
linear estimation of  
natural resonances in  
target indentification.

17 JAN 89  
11 NOV 89  
7 FEB 90

80019  
35,739  
35,739

Thesis  
C44887  
c.1

205817

Thesis  
C448872 Chong  
c.1

Investigation of non-  
linear estimation of  
natural resonances in  
target indentification.



thesC448872

Investigation of non-linear estimation o



3 2768 001 02657 8

DUDLEY KNOX LIBRARY